Revisit Overconfidence for OOD Detection: Reassigned Contrastive Learning with Adaptive Class-dependent Threshold

Yanan Wu[1*], **Keqing He**[2*], **Yuanmeng Yan**[1], **Qixiang Gao**[1], **Zhiyuan Zeng**[1], **Fujia Zheng**[1], **Lulu Zhao**[1], **Huixing Jiang**[2], **Wei Wu**[2], **Weiran Xu**[1*]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
{yanan.wu,yanyuanmeng,gqx,zengzhiyuan,fujia_zheng,
zhaoll,xuweiran}@bupt.edu.cn
{hekeqing,jianghuixing,wuwei30}@meituan.com

Abstract

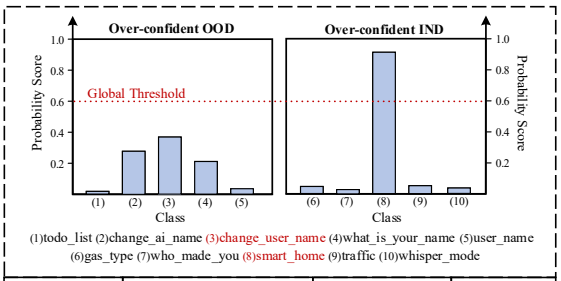
Detecting Out-of-Domain (OOD) or unknown intents from user queries is essential in a task-oriented dialog system. A key challenge of OOD detection is the overconfidence of neural models. In this paper, we comprehensively analyze overconfidence and classify it into two perspectives: over-confident OOD and in-domain (IND). Then according to intrinsic reasons, we respectively propose a novel reassigned contrastive learning (RCL) to discriminate IND intents for over-confident OOD and an adaptive class-dependent local threshold mechanism to separate similar IND and OOD intents for over-confident IND. Experiments and analyses show the effectiveness of our proposed method for both aspects of overconfidence issues.[1]

1 Introduction

Out-of-domain (OOD) detection is a key component of the task-oriented dialogue system(Gnewuch et al., 2017; Akasaki and Kaji, 2017; Shum et al., 2018; Tulshan and Dhage, 2019). It aims to decide whether a user query falls outside the range of predefined supported intents and avoid performing wrong operations (Lin and Xu, 2019; Xu et al., 2020; Zeng et al., 2021a). Due to the complexity of annotating OOD intents, most work focus on unsupervised OOD detection where there is no labeled OOD data but only labeled in-domain (IND) data (Xu et al., 2020). No prior knowledge about OOD intents makes it challenging to identify these unknown samples in the dialog system.

Existing unsupervised OOD detection methods mostly follow the same framework: firstly learn intent representations via labeled in-domain (IND) data then employ detecting algorithms, such as Maximum Softmax Probability (MSP) (Hendrycks and Gimpel, 2017), Local Outlier Factor (LOF) (Lin and Xu, 2019), Gaussian Discriminant Analysis (GDA) (Xu et al., 2020) to compute the similarity of features between OOD samples and IND samples. For example, Hendrycks and Gimpel (2017) simply uses a fixed threshold on the IND classifier's probability estimate and predicts a query as OOD only if its max logit is below the threshold. Lin and Xu (2019) employs an unsupervised density-based novelty detection algorithm, local outlier factor (LOF) to detect OOD intents. Further, Zeng et al. (2021a) proposes a supervised contrastive learning objective to learn discriminative intent features.

However, these methods ignore the key challenge of OOD detection, over-confidence. Guo et al. (2017); Liang et al. (2017, 2018) have theoretically proved that deep neural networks with the softmax classifier are prone to produce highly



Error Type	Input Sentence	Ground Truth	Baseline Predict
Over-confident OOD	can you call me a different name	change_user_name(IND)	OOD
Over-confident IND	please clean the car windows	OOD	smart_home(IND)

Figure 1: Examples of two kinds of overconfidence.

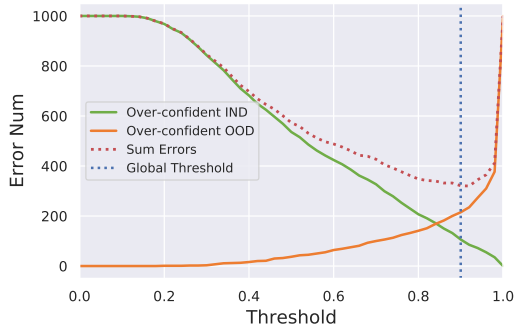


Figure 2: Number of error cases comparing over-confident IND and OOD. Global threshold denotes the best overall performance.

*The first two authors contribute equally. Weiran Xu is the corresponding author.

[1]We release our code at https://github.com/pris-nlp/NAACL2022-Reassigned_Contrastive_Learning_OOD.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4165 - 4179
July 10-15, 2022

over-confident posterior distributions even for such abnormal OOD samples. In this paper, we define the over-confidence issue from two aspects. (1) IND → OOD: Given an in-domain test sample, the pre-trained IND classifier predicts a lower confidence score than a fixed threshold[2] and incorrectly regards it as an OOD intent, which we name as **over-confident OOD**. We argue it's because IND classes have high semantic similarity then scatter and lower the max confidence score. For example, in Figure 1, the IND test query "can you call me a different name" is wrongly detected as OOD intent because its ground-truth IND label *change_user_name* is similar to the other IND categories *what_is_your_name* and *change_ai_name*. Thus the query gets comparable probability scores among the three IND classes, resulting in a lower max probability score than the threshold. (2) OOD → IND: Given an OOD test sample, the same classifier instead predicts a higher confidence score than the threshold and wrongly regards it as an IND, which we name as **over-confident IND**. The reason is the spurious correlation between OOD and IND intents, such as similar syntactic structure, entities, etc. For example, the OOD query "please clean the car windows" is classified into the IND category *smart_home* because plausibly similar examples like "please lock the doors" also exist in this category. The spurious correlation is frequent since humans define OOD without a clear and standard principle. Existing models (Lee et al., 2018; Ren et al., 2019; Zheng et al., 2020; Xu et al., 2020) mainly focus on the latter aspect over-confident IND, but we find the former over-confident OOD also makes a side effect on OOD detection. As Figure 2 shows, as the threshold rises, the number of negative OOD samples also increases which denotes the over-confident OOD issue gets worse but the over-confident IND issue gets better. We need to consider both two aspects of overconfidence.

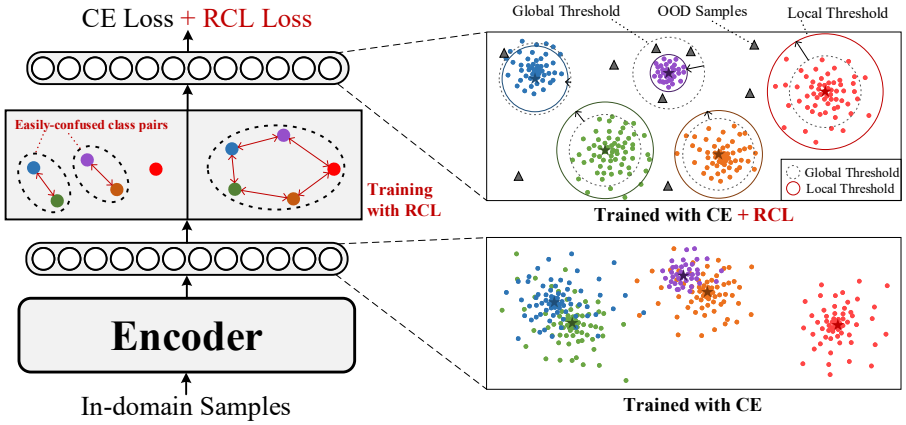
In this paper, we propose a novel reassigned contrastive learning (RCL) to discriminate intent representations between semantically similar IND categories and an adaptive class-dependent local threshold mechanism to separate similar IND and OOD intents. Specifically, for over-confident OOD, we first construct hard contrastive pairs among easily misclassified IND types using a pre-trained intent classifier. Then we train a new model to learn discriminative intent representations for similar IND categories via supervised contrastive learning (Khosla et al., 2020; Gunel et al., 2020). We aim to sample hard contrastive batches where anchors and positives have the same class label but different classifier outputs (hard positives), and anchors and negatives have the same classifier output but different class labels (hard negatives). For over-confident IND, we propose an adaptive class-dependent local threshold mechanism to separate similar IND and OOD intents. Traditional detection methods like MSP, GDA, Energy (Liu et al., 2020) use a global threshold to identify the confidence score of a test query, ignoring the difference between each IND class with OOD samples. We aim to adjust the class-dependent local threshold so that semantically correlated OOD and IND classes have a higher threshold to mitigate the model's overconfidence to IND.

Our contributions are three-fold: (1) We perform a comprehensive study on the overconfidence issue of OOD detection and analyze two-aspect reasons. (2) We propose a novel reassigned contrastive learning (RCL) to discriminate IND intents for over-confident OOD and an adaptive class-dependent local threshold mechanism to separate similar IND and OOD intents for over-confident IND. (3) Experiments and detailed analyses demonstrate the effectiveness of our proposed method for both aspects of overconfidence issues.

2 Related Work

OOD Detection Unsupervised models use only IND data for OOD detection following the threshold-based protocol, including modeling the probability density (Pidhorskyi et al., 2018), reconstruction (Golan and El-Yaniv, 2018), using classifier ensembles (Vyas et al., 2018; Shu et al., 2017), Bayesian models (Malinin and Gales, 2018), likelihood ratios (Ren et al., 2019). Note that all these methods require a dev set of labeled OOD intents to tune a fixed global threshold hyperparameter. We propose a more robust and efficient local threshold mechanism both to improve OOD performance and reduce the need for a large dev set of labeled OOD data. Another type of OOD detection model aims to utilize a set of OOD data in the training phase, including N+1 classifier (Fei and Liu, 2016; Zhan et al., 2021), entropy regularization (Zheng et al., 2020), adversarial augmentation (Zeng et al., 2021c).

[2]The threshold is tuned via the dev set. We will discuss it later.



a) The overall architecture b) The representation space

Figure 3: The overall architecture of our proposed approach. Reassigned contrastive learning (RCL) discriminates intent representations between semantically similar IND categories for over-confident OOD and adaptive class-dependent local threshold separates similar IND and OOD intents for over-confident IND.

Contrastive Learning Recent contrastive learning methods (Chen et al., 2020; He et al., 2020) have proven effective to learn unsupervised representations for downstream tasks. Winkens et al. (2020); Zeng et al. (2021b) combine cross-entropy loss on labeled IND data and instance-wise contrastive learning (CL) loss on unlabeled data (including unlabeled IND and OOD intents). They require a large amount of unlabeled corpus and can't explicitly distinguish different intent types. Further, Zeng et al. (2021a) uses supervised contrastive learning (SCL) (Khosla et al., 2020) to learn discriminative intent representations only using labeled IND data. Compared to CL, SCL regards all the IND intents from the same class as positive pairs and samples from different classes as negative pairs. However, we find intents within similar categories are still easily misclassified (see Section E). Thus, we propose a simple but strong reassigned contrastive learning (RCL) framework to give more penalty on these easily-confused IND classes to explicitly distinguish them. RCL aims to learn discriminative intent representations for OOD detection. Zhuang et al. (2019); Wang and Liu (2021) mines negatives close to the anchor sample as hard negatives by computing representation cosine similarity, but RCL uses the model's wrong predictions as supervised positives and negatives. Our method is more accurate because estimating representation similarity may be biased and we can construct both hard positives and negatives.

3 Methodology

Figure 3 shows the overall architecture of our proposed RCL and class-dependent local threshold where RCL discriminates easily-confused IND intents and local threshold separates similar IND and OOD intents. We follow a two-stage framework: first train an in-domain intent classifier in the training stage, then extract the intent feature of a test sample and employ the detection methods in the test stage.

3.1 Reassigned Contrastive Learning

Traditional models (Hendrycks and Gimpel, 2017) use cross-entropy (CE) loss to train an IND intent classifier which does not explicitly distinguish the margins between IND categories. Later, Lin and Xu (2019) and Zeng et al. (2021a) respectively propose a large margin cosine loss (LMCL) and a supervised contrastive learning (SCL) loss to minimize intra-class variance and maximize inter-class distance. However, we find IND intents within similar categories are still easily misclassified (see Section E). Therefore, we aim to give more penalty on these easily-confused IND classes to learn discriminative intent representations.

We first review the original contrastive learning (CL) and supervised contrastive learning (SCL) then introduce our RCL framework. Given an IND sample x_i and its intent label y_i, we adopt a BiLSTM (Hochreiter and Schmidhuber, 1997) or BERT (Devlin et al., 2019) encoder to get the intent representation z_i. Following Chen et al. (2020); Zeng et al. (2021b), we formulate CL loss for a positive pair of examples (i, j) as:

$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}\left(z_i, z_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}\left(z_i, z_k\right)/\tau\right)} \quad (1)$$

where $\mathbb{1}_{[k\neq i]} \in \{0, 1\}$ is an indicator function

evaluating to 1 if $k \neq i$. τ denotes a temperature parameter. The final loss is computed across all positive pairs, both (i, j) and (j, i) in a mini-batch of N examples. CL regards two augmented [3] views of a sample as positive pairs and views of different samples as negative pairs. Further, SCL extends the positive set by adding views of different samples from the same IND intent class and vice versa. However, these methods ignore easily-confused relations between semantically similar IND classes and can't separate them in the latent space. Therefore, we add more penalty to these easily-confused IND classes to learn discriminative intent representations. Specifically, our proposed reassigned contrastive learning (RCL) framework includes three stages.

IND classifier training First, we train an initial intent classification model M_{init} on the labeled IND dataset $\{(x_i, y_i)\}_{i=1}^{n}$ using CE, and save its predictions $\{\hat{y}_i\}_{i=1}^{n}$ on the training IND datapoints. We will use these outputs to train a more discriminated model.

Confused-label pair contrasting Here we aim to separate easily misclassified IND types. Thus we use the M_{init} outputs to obtain confused label pairs of the training data. For example, if class A and B have misclassified error cases, we use all the A and B's samples to construct contrastive batches. Then we perform SCL on these batches to train a new intent model M from scratch [4]. The intuition is that we treat samples that have the same class label but different classifier outputs as hard positives, and samples that have the same classifier output but different class labels as hard negatives. We display an example as Figure 4. Essentially, we restrict the new model to focus on misclassified intent classes by adding hard positives and negatives, further to learn discriminative intent representations. Different from existing hard CL work (Zhuang et al., 2019; Wang and Liu, 2021) which only consider close negatives as hard negatives using representation similarity, RCL uses the model's wrong predictions as supervised positives and negatives. Our method is more accurate because estimating representation similarity may be biased and we can construct both hard positives and negatives. We also perform SCL on other clean IND types.

[3] In this paper, we use adversarial augmentation as Zeng et al. (2021a). Please see more details in the original paper.

[4] We find training a new model M from scratch is much better than continual finetuning M_{init}. We argue it's hard to remove intrinsic knowledge existing in a model.

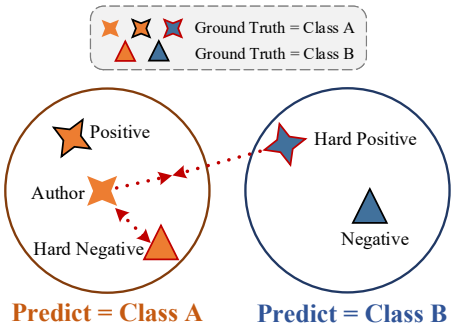


Figure 4: An example of confused label pair (A, B) where the same color denotes the same prediction class of the initial model.

In the experiments, we iteratively repeat the two processes for 5 epochs.

Global contrasting Apart from the confused label pairs, we also employ SCL on all the IND samples to avoid knowledge forgetting. Following Zeng et al. (2021a), we regard views of different samples from the same IND intent class as positives and views of different samples from the different IND intent class as negatives. Finally, we use CE to fine-tune the model M. In the experiments, we set the training epoch of SCL and CE to 10 and 20.

3.2 Adaptive Class-Dependent Local Threshold

Previous detection methods usually use a global threshold to identify the confidence score of a test query, ignoring the difference between individual IND classes with OOD samples. For example, if a test OOD query is similar to an IND type, it may obtain a high confidence score on this IND category and be wrongly regarded as IND. Therefore, we aim to set adaptive class-dependent local thresholds to avoid over-confident IND.

Previous methods using global threshold (Xu et al., 2020; Zeng et al., 2021a) first compute the max confidence scores of all the OOD and IND intents on the dev set, like max probability score, then adjust the threshold to maximize OOD F1 on the dev set. Notice it's a general and standard setting where a few labeled OOD data exists in the dev set for hyperparameter tuning for the OOD detection task.[5] For the local threshold, we input all OOD and IND queries to the pre-trained classifier and then get the confidence scores belonging to each IND type. Here we can use existing detection

[5] Section 5.4 shows our local threshold requires only 20 OOD intents on CLINC to achieve excellent metrics compared to 15, 000 IND intents in the training set, which is more robust and efficient.

	CLINC-Full		CLINC-Small		Snips	
	IND	OOD	IND	OOD	IND	OOD
Vocabulary size	6240		5725		11241	
Avg utterance length	9		9		9	
Intents	150	-	150	-	5	2
Training Set Size	15000	-	7600	-	9345	-
Samples per class	100	-	50	-	1869	-
Development Set Size	3000	100	3000	100	500	20
Samples per class	20	-	20	-	20	-
Testing Set Size	5500	1000	5500	1000	500	200
Samples per class	30	-	30	-	100	-

Table 1: Full statistics of OOD datasets, CLINC-Full, CLINC-Small and Snips.

methods to get confidence scores, such as MSP, GDA, Energy, etc. Taking MSP as an example, we get probability scores of all the intents. Then we group all the OOD samples into the corresponding IND type according to the max probability. So in each group, we have both OOD and IND intents and learn the class-dependent threshold by maximizing OOD F1 on each group. If no OOD sample is grouped into an IND type, we simply select the global threshold as the local threshold for this IND type. For inference, we select the local threshold of corresponding IND category where the test query gets the max probability score and predict it as OOD if the score is below the local threshold, Otherwise IND. Note that tuning local threshold doesn't increase too much computation cost but only multiple judgments and no extra cost in the inference. Local threshold achieves the best performance on CLINC and Snips even only using 20 OOD intents and is robust to different datasets and different number of OOD intents.

4 Experiments

4.1 Datasets

We perform experiments on three public benchmark OOD datasets, including CLINC-Full, CLINC-Small (Larson et al., 2019) and Snips (Coucke et al., 2018). We show the detailed statistic of these datasets in Table 1. Snips is a personal voice assistant dataset which contains 7 types of user intents across different domains. We randomly sample two classes among all classes in Snips, regarding them as OOD classes and the rest as IND classes. CLINC-Full and CLINC-Small both contain 150 IND intents across 10 domains. CLINC-Full has 100 training samples for each IND type, while CLINC-Small contains 50. We follow the standard dataset split Larson et al. (2019) and use the collected OOD test queries for evaluation. Note that all the datasets we used have a fixed set of labeled OOD data but we don't use it for training. We notice some work (Zhang et al., 2021) use a different split in CLINC-Full dataset where they sample 25%, 50%, 75% of all IND classes as IND, the other IND classes as OOD. The simulated split makes OOD data similar to IND data which class clusters are more compact thus get higher metrics. In this paper, we mainly follow the standard dataset split unless otherwise stated. For fair comparison, we also perform the same dataset split 25%, 50%, 75% in Table 3 and Table 10.

4.2 Metrics

We report both OOD metrics: Recall and F1-score (F1) and in-domain metrics: F1-score (F1). Since, we aims to improve the performance of detecting out-of-domain intents from user queries, OOD Recall and F1 are the main evaluation metrics in this paper.

4.3 Baselines

In training stage, we compare RCL with CE and SCL. In detection stage, we compare local threshold with global threshold. To verify the generalization of our proposed models, we use three OOD detection algorithms MSP (Maximum Softmax Probability)(Hendrycks and Gimpel, 2017), GDA (Gaussian Discriminant Analysis)(Xu et al., 2020) and Energy(Ouyang et al., 2021). Besides, we compare our models with the following state-of-the-art baselines, OpenMax(Bendale and Boult, 2016a), DeepUnk(Lin and Xu, 2019), Energy(Ouyang et al., 2021), SCL(Zeng et al., 2021a) and ADB(Zhang et al., 2021). We provide a more comprehensive comparison and implementation details of these models in the Appendix.

4.4 Implementation Details

To conduct a fair comparison, we follow a similar evaluation setting as (Zeng et al., 2021a) and (Zhang et al., 2021). We use the public pre-trained GloVe embeddings (Pennington et al., 2014) and BERT-uncased (Devlin et al., 2019) (with 12-layer transformer, implemented in PyTorch) to embed tokens. We set the learning rate to 1e-03 for LSTM and 2e-05 for BERT. To speed up the traning procedure and achieve better performance, we freeze all but the last transformer layer parameters of BERT. We use Adam optimizer (Kingma and Ba, 2014) to train our model and set the dropout rate to 0.5. We use the best F1 scores on the development set to calculate the MSP, GDA and Energy thresholds adaptively. Each result of the experiments is tested

Detection	Training	CLINC-Full						Snips					
		Global Threshold			Local Threshold(ours)			Global Threshold			Local Threshold(ours)		
		OOD		IND	OOD		IND	OOD		IND	OOD		IND
		F1	Recall	F1	F1	Recall	F1	F1	Recall	F1	F1	Recall	F1
MSP	CE	54.70	44.50	86.28	64.35	62.00	87.06	74.39	80.19	88.37	78.03	82.46	90.21
	SCL	56.98	46.34	87.94	65.88	64.31	88.00	76.23	80.57	89.60	79.25	82.57	91.30
	RCL(ours)	**61.71**	**53.90**	88.45	**67.43**	**64.92**	88.76	**81.00**	**81.52**	91.71	**83.53**	**82.94**	93.28
GDA	CE	65.79	64.14	87.90	68.06	73.50	87.95	77.33	79.23	90.08	81.96	84.52	90.11
	SCL	68.04	66.92	88.60	70.85	73.40	88.63	80.27	82.46	91.19	83.45	87.20	92.58
	RCL(ours)	**72.61**	**70.00**	88.98	**73.88**	**75.10**	89.03	**85.24**	**86.95**	93.89	**87.91**	**88.57**	94.65
Energy	CE	68.87	66.30	88.02	71.67	72.50	88.78	78.75	79.27	91.00	82.65	84.70	92.58
	SCL	71.12	71.01	88.59	73.15	73.20	88.98	81.72	81.99	91.27	85.04	85.83	93.47
	RCL(ours)	**74.30**	**72.03**	89.56	**75.32**	**78.60**	89.67	**86.41**	**87.16**	94.40	**89.21**	**89.45**	95.42

Table 2: The performance of OOD detection and IND classification with different OOD detection methods on CLINC-Full and Snips datasets for the BiLSTM-based model($p<0.05$ under t-test).

Models	CLINC-50%		CLINC-Full	
	OOD F1	IND F1	OOD F1	IND F1
OpenMax(Bendale and Boult, 2016a)	81.89	80.54	-	-
DeepUnk(Lin and Xu, 2019)	85.85	82.11	-	-
Energy(Ouyang et al., 2021)	84.34	82.61	75.93	91.23
SCL(Zeng et al., 2021a)	86.42	84.55	68.21	89.57
ADB(Zhang et al., 2021)	88.65	85.00	76.52	90.94
Ours	**92.16**	**86.05**	**82.03**	**92.00**

Table 3: The performance of OOD detection and IND classification compared with previous state-of-the-art baselines for the BERT-based model.

for 10 times under the same setting and reports the average value. In the training stage, for our model, we conduct 5 epochs of confused-label pair contrasting on designative batches of IND data, and then 10 epochs of global contrasting on randomly sampled IND data. Finally, we used CE to finetune the previous model with the epoch to 20. For the baselines in Table 2, we set the training epoch to 20 for CE and 15 for SCL. For fair comparison, we adopt the same data augmentation method as (Zeng et al., 2021a). Specifically, we apply adversarial attack to generate pseudo positive samples to increase the diversity of views for contrastive learning. The training time for CE is about 1.6 minutes using Glove+LSTM, and 12 minutes using BERT. The training stage of our model lasts about 2 minutes using Glove+LSTM, and 15 minutes using BERT on single Tesla T4 GPU (16 GB of memory) in CLINC-Full dataset which has 15,000 training samples. And the test stage of our model lasts about 1 second using Glove+LSTM, and 3 seconds using BERT. We have similar training time with SCL. And the test time of RCL is the same as that of CE and SCL.

4.5 Main Results

Table 2 displays our experimental results on datasets of CLINC-full and Snips with three different OOD detection algorithms: MSP, GDA and Energy. We show similar results on CLINC-Small dataset in the Appendix Table 7. From Table 2, we can make the following observations.

Our method achieves the best results under all detection algorithms. Using RCL+Local threshold significantly outperforms all the baselines under different OOD detection algorithms. Specifically, our method achieves 10.44%, 5.84%, 4.20% improvements over SCL+Global threshold on OOD F1 under three OOD detection algorithms on CLINC-full dataset and 7.30%, 7.64% and 7.49% on Snips dataset. We also observe that F1 score of IND classification with our approaches can keep comparable or slightly outperform the baselines, showing that RCL and class-dependent local threshold effectively improve OOD detection without harming the performance of IND classification. We show similar results on CLINC-Small dataset in the Appendix Table 7.

RCL consistently outperforms CE and SCL. On CLINC-Full, RCL achieves 4.73%, 4.57% and 4.18% improvements over SCL on OOD F1 under three OOD detection settings and 4.77%, 4.97% and 4.69% on Snips dataset, respectively. It demonstrates that RCL can stably discriminate the representation space by separating easily-confused classes and thus improve the performance of OOD detection. We also find compared with local threshold, RCL improved more significantly on global threshold. We argue this is due to the local threshold has alleviated part of overconfidence problem.

Local threshold consistently outperforms global threshold. We find the local threshold consistently wins the global one with a significant margin, especially in MSP. We argue MSP suffers from more serious over-confident IND issue than GDA and energy. This reveals that the local threshold can alleviate the over-confident IND issue by assigning an adaptive threshold for each IND class.

Comparing with previous baselines. To make a

Models	Min	Median	Max	Mean
CE	162.9	252.6	445.5	261.1
SCL	125.0	228.8	408.7	235.9
RCL(ours)	**96.4**	**177.9**	**390.2**	**190.6**

Table 4: Intra-class variance statistics.

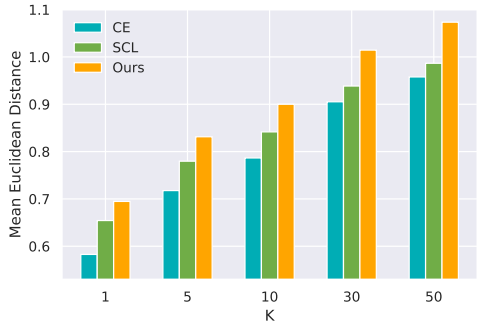


Figure 5: Inter-class distance statistics with different number of the nearest class centers.

fair comparison with previous baselines, we follow the same setting as Zhang et al. (2021). Specifically, we use the BERT as the backbone of the model, and sample 25%, 50%, 75% or 100% (Full) classes of the CLINC-Full dataset as the IND classes, the other as OOD. From Table 3 and 10 (in the Apendix), we can observe that our approach outperforms previous baselines on F1 scores of both OOD detection and IND classification. We achieve 4.56%, 3.51%, 4.03%, and 5.51% performance gain on OOD F1 for the 25%, 50%, 75% and Full settings.

5 Analysis

5.1 Analysis of IND Representations

To analyse how our RCL method affects the representation space and improves the performance of OOD detection, we perform a statistical analysis of the intra-class variance within each class as well as the inter-class distance between multiple classes.

RCL leads to smaller intra-class variance. To calculate the intra-class variance, we first compute the class center by averaging all samples representation corresponding to same class, then we calculate the variance of all samples to corresponding class center as the corresponding intra-class variance. In Table 4, we show the intra-class variance statistics among all classes with different training strategies: CE, SCL and our proposed RCL. We can find that all of the minimum, median, mean and maximum values of our approach are lower than CE and SCL, demonstrating that the RCL makes the representations within a single class tighter.

RCL leads to larger inter-class distance. We calculate the inter-class distance by averaging the euclidean distance from the center class to its K nearest classes. For each class, we take the class center by averaging all representations that belong to this class. Figure 5 show that our RCL approach consistently obtains larger inter-class distance compared to CE and SCL. This phenomena shows that our approach improves the OOD detection performance by separating among classes while maintaining intra-class high cohesion. It effectively improves the uniformity of the representation space.

Type 1	Type 2	Models	Inter-class Distance
change_user_name	change_ai_name	CE	0.526
		SCL	0.573
		Ours	**0.900**
change_user_name	payday	CE	1.023
		SCL	1.151
		Ours	**1.181**
what_is_your_name	user_name	CE	0.449
		SCL	0.466
		Ours	**0.789**
change_speed	play_music	CE	0.774
		SCL	0.925
		Ours	**1.094**
rewards_balance	redeem_rewards	CE	0.511
		SCL	0.572
		Ours	**0.666**

Table 5: Inter-class distance statistics between Confused Pairs.

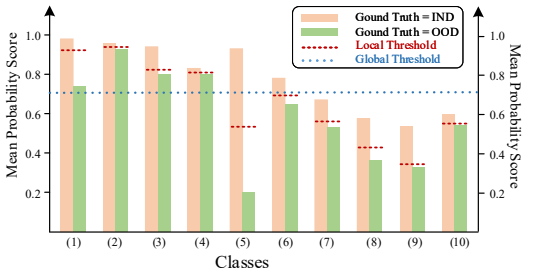
5.2 Analysis of Confusing IND Categories

To further verify the effect of RCL on easily-confused classes, we select 4 label pairs that are easily confused by the baseline model (*change user name* v.s. *change ai name*, *what is your name* v.s. *user name*, *change speed* v.s. *play music*, and *rewards balance* v.s. *redeem rewards*). In addition, we also select one label pair that is semantically unrelated (*change user name* v.s. *payday*). For each class pair, we compare the inter-class of models that are trained with CE, SCL and our RCL.

We can observe from the group that training with RCL greatly increases the inter-class distance between confusable classes in Table 5. Take the first class pair as an example, our RCL achieves 57% gain of inter-class distance over the SCL baseline, while the SCL only achieves 9% gain based on CE. On the other hand, when comparing between the group 1 and 2, we can conclude that our method works better on pushing away confusable class pairs, while for semantically unrelated ones, the effect is not obvious (+ 57% v.s. + 2.6%).

5.3 Effect of Local Threshold

As we discussed above, setting a general global threshold for OOD detection is a straightforward



(1)vaccines (2)credit_limit_change (3)travel_notification (4)account_blocked (5)order (6)timezone
(7)play_music (8)maybe (9) change_user_name (10)bill_balance

Figure 6: Local Threshold vs Global Threshold.

idea and the common practice in reality. However, it can not be applied to the situation where the variance of each IND class is various. If the global threshold is not suitable for some specific IND classes, the probability of true OOD samples may exceed the global threshold, resulting in over-confident IND samples.

To show the effect of class-dependent local threshold, we randomly select 20 IND classes, and give the mean softmax probability of the IND samples of each class as well as the OOD samples that have the same class with maximum softmax probability (i.e., these samples may be misclassified to this IND class or correctly identified as OOD, depending on the threshold). Then, we give the global threshold and the class-dependent local threshold for each IND class. For simplicity, we only show the mean value but variance. Note that the threshold between IND and OOD means most of the judgments are correct, but not necessarily all of them. We show half of results in Figure 6 and the rest in the Appendix(Figure 10).

For IND classes (1) - (4), the averaged maximum softmax probabilities of OOD samples are all beyond the threshold, indicating that a large proportion of the OOD samples are over-confident and will be classified to the corresponding IND class. However, with our class-dependent local threshold, the threshold is greater than the global threshold and can distinguish the IND and OOD samples more precisely.

For IND classes (5) - (6), the global threshold is lucky to distinguish the IND and OOD samples. However, we find these classes only make up a small proportion of all IND classes (about 20% since only 2 classes among 10 belong to this case).

For IND classes (7) - (10), we find even the averaged maximum softmax probability of IND samples is below the global threshold. It means that these IND classes can be easily confused with other

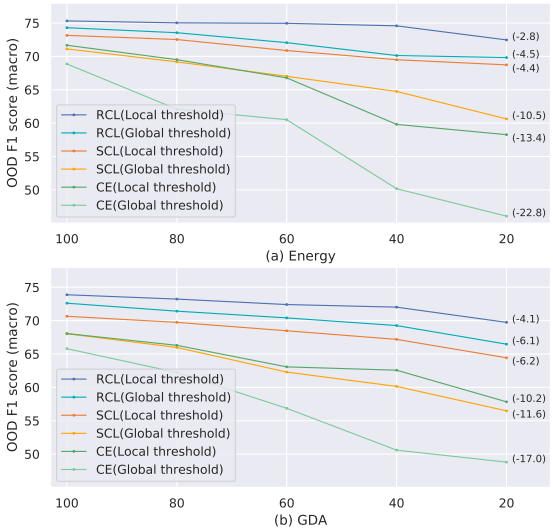


Figure 7: Effect of out-of-domain data size in development set.

semantically related IND classes, and thus degrades their maximum softmax probabilities, resulting in over-confident OOD samples. In contrast, our local threshold can automatically select the proper threshold, adaptively degrading the threshold correspondingly so that the threshold can exactly separate the IND and OOD samples.

5.4 Effect of Development Set Size

Figure 7 shows the effect of development set size. Using a small amount of OOD samples in development set to get a more suitable threshold is inevitable in OOD detection task. To show the effect of development set size, we randomly choose development data with a certain proportion from CLINC-Full OOD labeled development set and use the original test set for evaluation. We use the LSTM+Energy and LSTM+GDA settings.

Compare local and global thresholds. We find that the local threshold consistently outperforms global threshold with a significant margin, regardless of the OOD detection methods and training strategies we applied. Specifically, the performance drops are -4.5%(-2.8%), -10.5%(-4.4%) and -22.8%(-13.4%) over global(local) thresholds with three training strategies, respectively. Besides, we also find that the CE+Global collapses in performance when there are only 20 samples in dev set, while CE+Local is much better. We speculate that this phenomenon occurs because the OOD samples in the CLINC-Full dataset are diverse and the global threshold tends to overfit the data. It confirms that our proposed local threshold method can alleviate the reliance on development size by

Models	IND<->OOD			IND<->IND	All Errors
	IND->OOD	OOD->IND	SUM		
MSP	156	576	732	290	1022
MSP + RCL	**122**	542	664	233	897
MSP + Local Threshold	284	**378**	662	234	896
MSP + RCL + Local Threshold	229	390	**619**	**194**	**813**

Table 6: Statistics of different error types using RCL and Local Threshold.

assigning a specific threshold for each class.

Compare RCL with other training strategies. We find that the RCL consistently outperforms SCL and CE. Specifically, under CE, SCL and RCL, the performance drops are -22.8%, -10.5% and -4.5%, respectively. Besides, with the decrease of development data size, the corresponding difference gradually increases. It demonstrates that RCL can effectively relieve the dependence on development size by separating easily misclassified classes.

We also observe that in the range of 40-100, the performance of RCL hardly degrades. To be more specific, when only 40 OOD labeled data provided, the OOD F1 score of RCL+Local threshold model is still 74.58(-0.74%). And even when only 20 OOD labeled data are available(about 0.13% training data), our proposed model still outperforms the best baseline(SCL+Global threshold) with 100 OOD labeled data by 1.35%. This reveals that our model is more robust and less dependent on development set size. And the result of our model is significantly improved under four datasets further proves that our proposed methods have strong robustness and generalization capability.

5.5 Analysis of Error Types

In order to explore the effect of our method on different error types. We divide all the error samples into three categories: **confusion between IND**(IND<->IND), **over-confident OOD**(IND<->OOD) and **over-confident IND**(OOD<->IND).

Figure 8(a) indicates that RCL can achieve consistently improvements in all the three error types. Figure 8(b) indicates that local threshold outperforms global threshold both in error of IND<->IND and OOD->IND. While greatly reducing OOD->IND errors, local threshold inevitably increases IND->OOD errors (explained by Figure 2). We hypothesize this is due to the high semantic similarity between IND classes and labeling noise in dataset, which can be mitigated by RCL(see Table 6). We will leave more possible solutions for future work. Combined with Table 6, compared to global threshold, our local threshold can obtain smaller overall errors. In general, local threshold is better

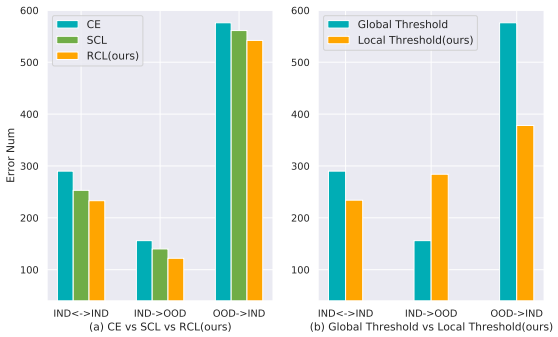


Figure 8: Statistics of different error types using different training objectives and different thresholds.

for the overall performance.

Table 6 displays the comparison between RCL and local threshold. We find our RCL and local threshold both outperform MSP baseline and RCL+local threshold achieves the best performance. Comparing RCL and local threshold, RCL targets at IND->OOD errors (from 156 to 122) for over-confident OOD, while local threshold helps reduce OOD->IND errors (from 576 to 378) for over-confident IND. Besides, they both help reduce overall IND&OOD errors. On the other hand, the performance improvement on OOD detection always helps IND classification.

6 Conclusion

In this paper, we focus on the overconfidence issue of unsupervised OOD detection. We find the reasons for overconfidence arise from two aspects: (1) IND classes have high semantic similarity. (2) IND and OOD intents have spurious correlations. According to the two reasons, we respectively propose a novel reassigned contrastive learning (RCL) to discriminate IND intents for over-confident OOD and an adaptive class-dependent local threshold mechanism to separate similar IND and OOD intents for over-confident IND. We perform extensive experiments and comprehensive analyses to demonstrate the effectiveness of our approach. We hope to provide new guidance for future work.

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions. This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

References

Satoshi Akasaki and Nobuhiro Kaji. 2017. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. *ArXiv*, abs/1705.00746.

Abhijit Bendale and Terrance E Boult. 2016a. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572.

Abhijit Bendale and Terrance E. Boult. 2016b. Towards open set deep networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 506–514, San Diego, California. Association for Computational Linguistics.

Ulrich Gnewuch, S. Morana, and A. Maedche. 2017. Towards designing cooperative and social conversational agents for customer service. In *ICIS*.

Izhak Golan and Ran El-Yaniv. 2018. Deep anomaly detection using geometric transformations. In *NeurIPS*.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *ArXiv*, abs/2011.01403.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *ICML*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ArXiv*, abs/1610.02136.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, A. Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *ArXiv*, abs/2004.11362.

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

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *ArXiv*, abs/1711.09325.

Shiyu Liang, Yixuan Li, and R. Srikant. 2017. Principled detection of out-of-distribution examples in neural networks. *ArXiv*, abs/1706.02690.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv: Learning*.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.

Weitang Liu, Xiaoyun Wang, John Douglas Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *ArXiv*, abs/2010.03759.

Andrey Malinin and Mark John Francis Gales. 2018. Predictive uncertainty estimation via prior networks. In *NeurIPS*.

Yawen Ouyang, Jiasheng Ye, Yu Chen, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Energy-based unknown intent detection with data manipulation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2852–2861, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*

(EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Stanislav Pidhorskyi, Ranya Almohsen, Donald A. Adjeroh, and Gianfranco Doretto. 2018. Generative probabilistic novelty detection with adversarial autoencoders. In *NeurIPS*.

J. Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. In *NeurIPS*.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *EMNLP*.

H. Shum, X. He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19:10–26.

Amrita S. Tulshan and Sudhir N. Dhage. 2019. Survey on virtual assistant: Google assistant, siri, cortana, alexa. *Communications in Computer and Information Science*.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. 2018. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *ECCV*.

Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504.

J. Winkens, R. Bunel, Abhijit Guha Roy, Robert Stanforth, V. Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, T. cemgil, S. Eslami, and O. Ronneberger. 2020. Contrastive training for improved out-of-distribution detection. *ArXiv*, abs/2007.05566.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1452–1460, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021a. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. 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)*, pages 870–878, Online. Association for Computational Linguistics.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Hong Xu, and Weiran Xu. 2021b. Adversarial self-supervised learning for out-of-domain detection. In *NAACL*.

Zhiyuan Zeng, Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2021c. Adversarial generative distance-based classifier for robust out-of-domain detection. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7658–7662.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Y.S. Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3521–3532, Online. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *AAAI*.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1198–1209.

Chengxu Zhuang, Alex Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6001–6011.

A Comparison of CL, SCL and RCL

Contrastive learning (CL) methods (Chen et al., 2020; He et al., 2020) have been proven effective to learn unsupervised representations for downstream tasks. Winkens et al. (2020); Zeng et al. (2021b) propose to apply contrastive learning (CL) to OOD detection task. They first perform CL on unlabeled data (including unlabeled IND and OOD intents) to learn OOD representations, then use cross-entropy loss on labeled IND data to learn an IND intent classifier. But the unlabeled data is not always available. More importantly, CL can only independently learn the OOD and IND representations (because the CL loss is built in an instance-wise way), but not explicitly distinguish different intent types in a class-wise way. Further, Zeng et al. (2021a) uses supervised contrastive learning (SCL) (Khosla et al., 2020) to learn discriminative intent representations only using labeled IND data. Compared to CL, SCL regards all the IND intents from the same class as positive pairs and samples from different classes as negative pairs. SCL aims to learn tight intent representations for each intent type and tries to distinguish different intent types. However, we find intents within similar categories

Detection	Training	Global Threshold			Local Threshold(ours)		
		OOD		IND	OOD		IND
		F1	Recall	F1	F1	Recall	F1
MSP	CE	48.89	37.40	83.94	62.94	61.42	85.61
	SCL	55.79	47.20	86.25	63.57	61.10	86.80
	RCL(ours)	**61.54**	**53.23**	86.26	**65.33**	**62.05**	86.81
GDA	CE	61.89	60.72	84.87	63.88	66.60	84.40
	SCL	64.70	64.16	87.02	66.64	71.69	87.17
	RCL(ours)	**70.56**	**68.92**	87.27	**71.69**	**72.80**	87.54
Energy	CE	63.57	58.10	84.70	66.63	66.69	84.74
	SCL	67.64	65.10	85.77	70.67	70.16	86.89
	RCL(ours)	**71.46**	**71.49**	**87.69**	**74.19**	**77.90**	**87.86**

Table 7: The performance of OOD detection and IND classification with different OOD detection methods on CLINC-Small dataset (p<0.05 under t-test).

	Training	Test
SCL	x1.00	-
RCL(ours)	x1.10	-

Table 8: The radio of training time and test time compared with SCL model.

	Training	Test
Global Threshold	-	x1.00
Local Threshold(ours)	-	x1.06

Table 9: The radio of training time and test time compared with global threshold.

Models	CLINC-25%		CLINC-75%	
	OOD F1	IND F1	OOD F1	IND F1
OpenMax(Bendale and Boult, 2016a)	75.76	61.62	76.35	73.13
DeepUnk(Lin and Xu, 2019)	87.33	70.73	81.15	86.27
Energy(Ouyang et al., 2021)	91.09	72.68	71.43	78.07
SCL(Zeng et al., 2021a)	93.40	77.16	73.98	86.89
ADB(Zhang et al., 2021)	91.84	76.80	83.92	88.58
Ours	**96.40**	**83.56**	**87.95**	**89.67**

Table 10: The performance of OOD detection and IND classification compared with previous state-of-the-art baselines for the BERT-based model.

are still easily misclassified (see Section E). Intuitively, these easily-confused IND classes can be regarded as hard examples (Zhuang et al., 2019) for existing CL-based methods. Inspired by the idea, we propose a simple but strong reassigned contrastive learning (RCL) framework to give more penalty on these easily-confused IND classes to explicitly distinguish them. The main difference is the confused-label pair contrasting process (we provide an example to show how to construct hard positives and negatives in Figure 4) and please see details in the Confused-label pair contrasting section(line 241-267). Generally, RCL aims to learn discriminative intent representations for OOD detection, especially for these easily-confused intent classes. Compared to Zhuang et al. (2019); Wang and Liu (2021), these work only mines negatives close to the anchor sample as hard negatives by computing representation cosine similarity, but RCL uses the model's wrong predictions as supervised positives and negatives. Our method is more accurate because estimating representation similarity may be biased and we can construct both hard positives and negatives. Note that RCL only increases the training cost for a little but requires no extra inference budget and uses the same model size as SCL. Please see the following section for details.

B Comparison of time complexity and space complexity

We discuss the time and space complexity of RCL and Local threshold in Table 8 and Table 9. In terms of time complexity, we set the epoch of SCL and RCL to 15. The training time for RCL is about 15 minutes using BERT in CLINC-Full dataset which has 15,000 training samples. RCL have similar training time with SCL. And the test stage of global threshold lasts about 3 seconds using BERT. The local threshold has almost the same test time as the global threshold, while the performance can even be increased by 9%. From the perspective of space complexity, RCL and SCL utilize the same encoder structure. The size of the model parameters of RCL is equal to that of SCL. Besides, the global threshold and the local threshold are only the differences of the algorithm, and there are no extra parameters.

C Algorithm

We show the training procedure of RCL in Algorithm 1. E_0, E_1 and E_2 are the training epochs of confused-label pair contrasting, global contrasting and cross-entropy classification processes, respectively. In practice, we set E_0, E_1 and E_2 to 5, 10 and 20, respectively. n is the number of training samples. First, we construct a set of confused la-

Algorithm 1 : Reassigned Contrastive Learning

Input: training dataset $D_0 = \{(x_i, y_i)\}_{i=1}^n$, Batch size N, training epoch E_0, E_1 and E_2, initial intent classification model's predictions $D_1 = \{(x_i, \hat{y}_i)\}_{i=1}^n$

Output: a new intent classification model

construct confused label pairs set $P = \{(y_j, \hat{y}_j)|y_j \in D_0, \hat{y}_j \in D_1, y_j \neq \hat{y}_j\}_{j=1}^m$ and clean labels set $S = \{y_j|y_j \notin P\}_{j=1}^k$

for epoch = 1 to E_0 **do** ▷ Confused-label pair contrasting

  sample confused mini-batch $\hat{B} = \{\{(x_i, y_j \text{ or } \hat{y}_j))|(y_j, \hat{y}_j) \in P\}_{i=1}^N\}_{j=1}^m$ from D_0

  sample clean mini-batch $B = \{(x_i, y_i)|y_i \in S\}_{i=1}^N$ from D_0

  iteratively compute supervised contrastive loss on $\hat{B}$ or B

end for

for epoch = 1 to E_1 **do** ▷ Global contrasting

  random sample batches $B_1 = \{(x_i, y_i)\}_{i=1}^N$ from D_0

  compute supervised contrastive loss

end for

for epoch = 1 to E_2 **do**

  random sample batches $B_2 = \{(x_i, y_i)\}_{i=1}^N$ from D_0

  compute cross-entropy loss

end for

bel pairs by combining the ground truth labels y and predicted labels $\hat{y}$ of the training data. Taking Figure 4 as an example, (A, B) is one of confusing label pairs in P. m is the number of confusing label pairs. And the remaining labels that never confused with other labels, called clean labels, are collected in the set S. k is the number of clean labels. Then, we sample confused mini-batch $\hat{B}$ following P restrainedly. Note that a mini-batch $\hat{B}_j$ will only contain the samples with ground truth y_j or $\hat{y}_j$. We also sample clean mini-batch B following S restrainedly. Different from the confused mini-batch, a clean mini-batch B_j can contain any sample whose label belongs to S. We will compute the supervised contrastive loss iteratively, on confused mini-batch or clean mini-batch. Apart from confused-label pair contrasting, we also employ global contrasting by randomly sampling batches on all IND samples to avoid knowledge forgetting. Finally, we compute cross-entropy loss to fine-tune the model.

D Baselines

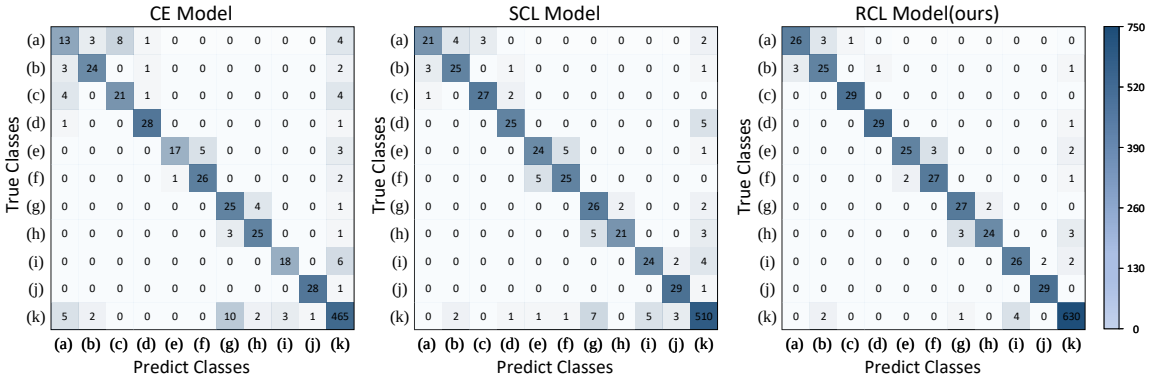
We compare many types of unsupervised OOD detection models. For feature extractor, we use LSTM(Long Short Term Memory)(Hochreiter and Schmidhuber, 1997) or BERT(Bidirectional Encoder Representations from Transformers)(Devlin et al., 2019). For training objection, we compare RCL with CE and SCL. For detection algorithms, to verify the generalization of our proposed models, we use MSP(Maximum Softmax Probability)(Hendrycks and Gimpel, 2017), GDA(Gaussian Discriminant Analysis)(Xu et al., 2020) and Energy(Ouyang et al., 2021). Besides, we compare our models with the following state-of-the-art baselines, OpenMax(Bendale and Boult, 2016a), DeepUnk(Lin and Xu, 2019), Energy(Ouyang et al., 2021), SCL(Zeng et al., 2021a), ADB(Zhang et al., 2021). We supplement the relevant baseline details as follows:

MSP (Maximum Softmax Probability)(Hendrycks and Gimpel, 2017) applies a threshold on the maximum softmax probability. We use the best F1 scores on the validation set to calculate the threshold adaptively.

GDA (Gaussian Discriminant Analysis)(Xu et al., 2020) is a generative distance-based classifier for out-of-domain detection with Euclidean space. It estimates the class-conditional distribution on feature spaces of DNNs via Gaussian discriminant analysis to avoid over-confidence problems and use Mahalanobis distance to measure the confidence score of whether a test sample belongs to OOD.

Energy(Ouyang et al., 2021) maps a sample $\mathbf{x}$ to a single scalar called the *energy*. We use the threshold on the energy score to consider whether a test query belongs to OOD.

OpenMax(Bendale and Boult, 2016b) is an open set detection method in computer vision, we adapt



(a)change_user_name (b)user_name (c)change_ai_name (d)what_is_your_name (e)redeem_rewards
(f)rewards_balance (g)ingredients_list (h)recipe (i)play_music (g)change_speed (k)unseen

Figure 9: Confusion matrix of CE, SCL and RCL based on MSP.

it for OOD. We firstly use the CE loss to train a classifier on in-domain intents, then fit a Weibull distribution to the classifier's output logits.

DeepUnk(Lin and Xu, 2019) learns the deep intent features with the margin loss and detects the unknown intent with local outlier factor.

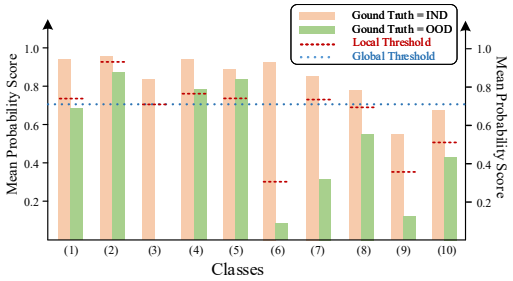
SCL(Zeng et al., 2021a) uses a supervised contrastive learning objective to learn discriminative intent features. We conduct many experiments and from multiple perspectives prove that our method can consistently outperform SCL.

ADB(Zhang et al., 2021) learns adaptive decision boundary using a loss function to balance both the empirical risk and the open space risk. It is still a time-consuming process. Compare with ADB, our method can achieve the best performance.

E IND Confusion Matrix

To demonstrate how our proposed RCL approach improves the performance by decreasing the number of error cases, we show the confusion matrix among 10 IND classes as well as the unseen OOD class. Specially, we pick 10 easily-confused classes by the baseline model as the IND classes, and show the confusion matrices of CE, SCL and RCL in Figure 9. From the figure, we can make the following observations.

On the one hand, we find the confusion of easily-confused IND classes is significantly mitigated. Take the beginning four IND classes *change user name* (a), *user name* (b), *change ai name* (c) and *what is your name* (d) as an example, when training with CE, there are totally 34 misclassified samples. However, after applying our RCL approach, the number misclassified samples decreases to 11, which is a 68% reduction. It indicates that the confusion among those semantically close IND classes



(1)calories (2)interest_rate (3)pto_balance (4)uber (5)update_playlist
(6)order_checks (7)definition (8)timezone (9)distance (10)restaurant_reviews

Figure 10: Local Threshold vs Global Threshold.

is effectively alleviated by our RCL training strategy.

On the other hand, we focus on the confusion between the unseen OOD and the IND classes. The last row of the confusion matrix indicates the over-confident IND (truly OOD sample, but predicted as IND), while the last column of the confusion matrix indicates the over-confident OOD (truly IND sample, but predicted as OOD). We can find that our RCL model reduces the number of over-confident IND samples by 69.6% and 70.0% compared to the CE and SCL models, respectively. Meanwhile, our RCL model also reduces the number of over-confident OOD samples by 60.0% and 47.4% compared to the CE and SCL models, respectively. This phenomena proves that our RCL model improves the OOD detection performance through solving the over-confident IND and OOD issues.

F Effect of Local Threshold

Figure 10 shows the result of the additional 10 IND classes. Limited by the degree of semantic similarity between OOD and IND, there may be no OOD samples on some classes. In this case, we will set the local threshold to be the same as the global threshold, as IND class (3).

For IND classes (1), (6), (7) and (8), both the

global threshold and local threshold are between the mean softmax probability of the IND samples and OOD samples. Analyze with specific examples, we find that the probability of OOD samples in (6) is more concentrated on small probability values, while (7) and (8) have greater variance. Compared with the global threshold, the local threshold can better alleviate the over-confident problem by choosing a more appropriate threshold boundary.

For IND classes (2), (4) and (5), the mean maximum softmax probabilities of OOD samples are all beyond the global threshold. For category (2), the local threshold falls exactly between IND and OOD while global does not. For (4) and (5), although both the local threshold and the global threshold are below OOD, it is clear that the local threshold is more reasonable.

For IND classes (9) and (10), both the averaged max softmax probability of IND and OOD are lower than the global threshold, which means that a large portion of IND samples will be misclassified as OOD. On the contrary, our local threshold method can adaptively select more appropriate thresholds, which largely eliminates the problem of overconfident OOD.