KNN-Contrastive Learning for Out-of-Domain Intent Classification

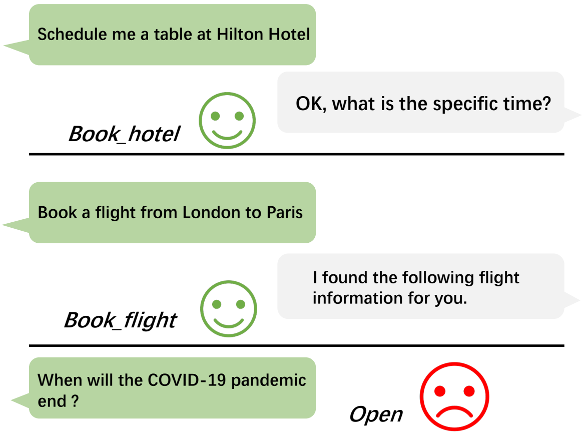
Yunhua Zhou[1,3], Peiju Liu[1,3], Xipeng Qiu[1,2,*]
[1]School of Computer Science, Fudan University
[2]Peng Cheng Laboratory
[3]Shanghai Collaborative Innovation Center of Intelligent Visual Computing
`{zhouyh20,xpqiu}@fudan.edu.cn`
`pjliu21@m.fudan.edu.cn`

Abstract

The Out-of-Domain (OOD) intent classification is a basic and challenging task for dialogue systems. Previous methods commonly restrict the region (in feature space) of In-domain (IND) intent features to be *compact* or *simply-connected* implicitly, which assumes no OOD intents reside, to learn discriminative semantic features. Then the distribution of the IND intent features is often assumed to obey a hypothetical distribution (Gaussian mostly) and samples outside this distribution are regarded as OOD samples. In this paper, we start from the nature of OOD intent classification and explore its optimization objective. We further propose a simple yet effective method, named KNN-contrastive learning. Our approach utilizes K-Nearest Neighbors (KNN) of IND intents to learn discriminative semantic features that are more conducive to OOD detection. Notably, the density-based novelty detection algorithm is so well-grounded in the essence of our method that it is reasonable to use it as the OOD detection algorithm without making any requirements for the feature distribution. Extensive experiments on four public datasets show that our approach can not only enhance the OOD detection performance substantially but also improve the IND intent classification while requiring no restrictions on feature distribution. Code is available.[1]

1 Introduction

People are getting accustomed to talking with or sending some instructions to task-oriented dialog system in natural language to assist them in fulfilling work. As the environment facing the dialogue systems becomes more open, there are more utterances with unknown or Out-of-Domain (OOD) intents that the dialog system does not know how to handle. As shown in Figure 1, the chatbot encounters an unsupported intent (OOD intent) utterance in the third interaction. It is significant to distinguish these utterances for the dialogue system because identifying the intents of the user determines whether subsequent actions can be carried out correctly.


Schedule me a table at Hilton Hotel

OK, what is the specific time?

Book a flight from London to Paris

I found the following flight information for you.

When will the COVID-19 pandemic end ?

Figure 1: An example about interactions between a user (green) and a chatbot (grey). The chatbot correctly understands the intents (In-domain intents) of the user in the first two rounds. Thirdly, the chatbot encounters Out-of-Domain intent and does not know how to do it.

To solve this problem, the existing methods roughly can be summarized into two categories according to whether extensive labeled OOD intent samples are used during training. The first kind of method (use OOD samples during training) is represented by (Zheng et al., 2020; Zhan et al., 2021; Choi et al., 2021) which regards OOD intent classification as a (n+1)-class classification task that the extra $(n+1)^{th}$ class represents labeled OOD intent. These methods may need additional large and time-consuming labeled Out-of-Domain samples. Moreover, manually constructed OOD samples endowed with artificial inductive bias cannot cover all open classes in the actual environment so this kind of method have their limitations.

In this paper, we focus on another kind of method which involves two stages, to learn dis-

[*] Corresponding author.
[1] https://github.com/zyh190507/KnnContrastiveForOOD.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5129 - 5141
May 22-27, 2022 ©2022 Association for Computational Linguistics

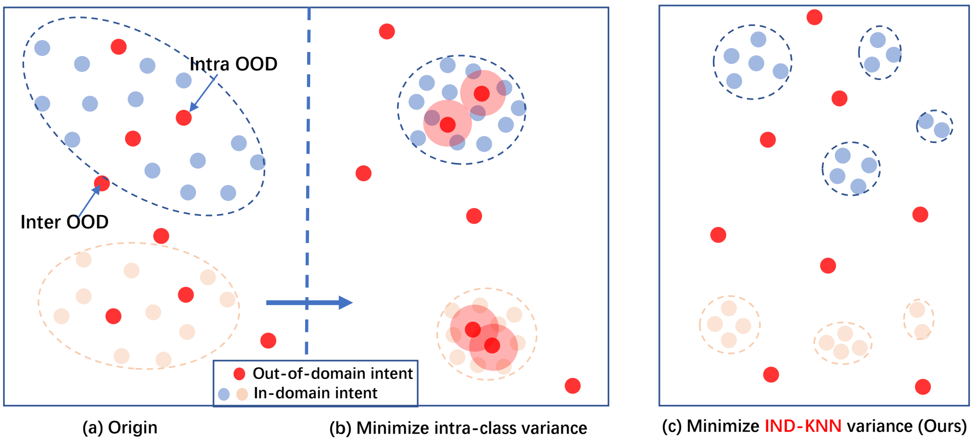

Intra OOD
Inter OOD
Out-of-domain intent
In-domain intent
(a) Origin
(b) Minimize intra-class variance
(c) Minimize IND-KNN variance (Ours)

Figure 2: (a) The original distribution of IND and OOD distribution. (b) Minimize the intra-class variance may increase the risk that OOD intents surrounded by a covex hull composed composed of IND samples locally are identified (red shadow) as IND. (c) Utilizes k-nearest neighbors of IND to contrastive learning.

criminative semantic features and OOD detection. **How to learn semantic features benefit for OOD detection?** Minimizing intra-class variance and maximizing inter-class variance, whose motivation are to facilitate detection by widening margin between IND and OOD intents, have always been regarded as the essence of solving this problem (Lin and Xu, 2019; Zeng et al., 2021a). Then, Shu et al. (2017); Zhang et al. (2021); Xu et al. (2020); Zeng et al. (2021a) impose (or implicitly) Gaussian distribution into the distribution of the learned intent features to OOD detection.

In general previous methods implicitly assume the region of semantic features as *compact* or *simply-connected region* in feature space, which means OOD intents only exist between different IND classes and not within IND distribution so that tight IND semantic features can be helpful for OOD detection. However, as shown in Figure 2 (a), the actual location of OOD intents in semantic space is not limited, they can appear between IND classes or within IND distribution. We name those OOD intents that are distributed between different IND classes as inter OOD intents, and those within IND distribution or can be surrounded by a convex hull composed of local IND intent samples as intra OOD intents. As shown in Figure 2 (b), for inter OOD intents, minimizing intra-class variance and maximizing inter-class variance can reduce the risk of being identified IND while this risk may be increased for intra OOD intents due to being closer to IND intents.

At the same time, we conduct Gaussian Hypothesis Testing on the IND semantic features distribution in the CLINC-FULL train set, which is learned by (Zeng et al., 2021a). we find only 57% IND classes conform to Gaussian distribution, which illustrates the Gaussian assumption for OOD detection in the previous methods may not be reasonable, see Appendix A.1 for results on other datasets and details.

To solve these problems, we explicitly define the optimization objective of OOD intents classification using open space risk (Scheirer et al., 2012). Compared with the previous methods only considering inter OOD intents, we propose a simple yet effective method to consider both intra and inter OOD intents. We utilize k-nearest neighbors of IND intent sample as positive samples as shown in Figure 2 (c), and obtain more negative samples with the help of queue in MoCo (He et al., 2020) to learn discriminative semantic features. We further analyze why our method can better reduce open space risk. Intuitively, our method leaves more margin around OOD intents, which can ensure we employ the basic density-based method for OOD detection without any assumptions about the distribution.

We summarize our contributions as follows. Firstly, following open space risk, we explicit the optimization objective of OOD intent classification to provide a paradigm for solving OOD intent classification. Secondly, we analyze the limitation of existing methods and propose a novel method to better reduce both empirical risk and open space risk. Thirdly, extensive experiments conducted on four challenging datasets show our approach achieves consistent improvements without restrictions on feature distribution.

2 Related Work

2.1 Out-of-domain Detection

Schölkopf et al. (2001); Tax and Duin (2004) regard Out-of-Domain detection as One-class classification problem so that to find a hyperplane or hypersphere by kernels in high-dimensional space to distinguish OOD, Scheirer et al. (2012) firstly proposes the definition of open space risk and formalize open set recognition as a constrained optimization problem. Then to obtain better semantic representation, deep neural network has been brought into this field in recent years. Bendale and Boult (2016) propose OpenMax model, using the scores from the penultimate layer of deep network, to distinguish OOD. MSP (Hendrycks and Gimpel, 2017) presents a baseline based on maximum softmax probabilities to exhibit the ability of the network to distinguish between OOD and IND. To enlarge the difference between IND and OOD, Liang et al. (2018) add temperature scaling based on MSP and adds perturbations to the inputs.

The above methods mainly focus on computer vision and assume (or implicitly) the feature region is compact (simply-connected region), many research works are carried out in natural language processing. DOC (Shu et al., 2017) builds a multi-class classifier and selects a threshold to reject. Further, they reduces the open space risk for rejection by tightening the decision boundaries of sigmoid functions with Gaussian fitting. LMCL (Lin and Xu, 2019) learns discriminative deep features with margin loss. Yan et al. (2020); Wan et al. (2018) model embeddings with a Gaussian mixture distribution to facilitate downstream outlier detection. Xu et al. (2020); Zeng et al. (2021b); Podolskiy et al. (2021) assume the IND semantic feature distribution as Gaussian discriminant analysis (GDA) and identify Out-of-domain samples by Mahalanobis distances. (Fei and Liu, 2016) reduce open space risk by decision boundaries and ABD (Zhang et al., 2021) propose to learn adaptive circular decision boundaries. Very recently, Zeng et al. (2021a) propose a supervised contrastive learning objective to maximize inter-class variance and to minimize intra-class variance. These methods also restrict the feature distribution in the feature learning stage or downstream detection stage and fail to solve Out-of-domain classification completely.

2.2 Contrastive Learning

Contrastive learning, which can be traced back to (Hadsell et al., 2006), is widely used in unsupervised or self-supervised learning (He et al., 2020; Wang and Isola, 2020; Khosla et al., 2020). With similarity by dot product, Gutmann and Hyvärinen (2010) propose InfoNCE loss to measure the similarities of sample pairs in semantic space. To obtain more number of negative samples for contrastive learning, He et al. (2020) introduce Momentum Contrastive (MoCo) that builds a large and consistent dictionary facilitating contrastive unsupervised learning. With the prevalence of pre-trained models (PTMs) (Qiu et al., 2020; Lin et al., 2021) in different fields, Dwibedi et al. (2021); Li et al. (2021) combine PTMs with contrastive learning paradigm, which adopts neighbors and uses MoCo or Memory Banks to obtain enough negative samples.

3 Proposed Method

3.1 Objective of OOD Intent Classification

Open space risk We define open space $\mathcal{O}$ and open space risk as follow (Scheirer et al., 2012; Bendale and Boult, 2015). Using IND training samples, open space can be defined as [2]:

$$\mathcal{O} = \mathcal{S} - \bigcup_{x \in \mathcal{X}} \sigma(x), \tag{1}$$

where σ is a local (and small) semantic space spanned by IND training sample x, $\mathcal{X}$ is the set of all IND training samples and $\mathcal{S}$ including both open space $\mathcal{O}$ and remaining space. Consider a measurable recognizer (or discriminator) f that $f(x) = 1 (> 0)$ for the IND intents, otherwise $f(x) = 0 (<= 0)$, probabilistic open space risk $\mathcal{R}_{\mathcal{O}}$ can be formed in terms of Lebesgue measure:

$$\mathcal{R}_{\mathcal{O}}(f) = \frac{\int_{\mathcal{O}} f(x,\theta) dx}{\int_{\mathcal{S}} f(x,\theta) dx}. \tag{2}$$

Objective of OOD Intent Classification To identify OOD intents, we need to learn intent representations at first, which also ensures the classification quality of IND in addition to adapting to downstream detection. Therefore we introduce an additional optimization objective as also suggested in (Scheirer et al., 2012; Bendale and Boult, 2015),

[2] We also define a specific open space risk for limited OOD samples. See the Appendix A.6 for details and more discussion.

named empirical risk $\mathcal{R}_\varepsilon(f)$. The objective can be defined as:

$$\underset{f(x,\theta)\in\mathcal{H}}{\arg\min}\{(1-\lambda)\cdot\mathcal{R}_\varepsilon(f)+\lambda\cdot\mathcal{R}_\mathcal{O}(f)\}, \quad (3)$$

where λ is a hyper-parameter to balance empirical and open space risk and $\mathcal{H}$ is the function space.

3.2 Minimize Empirical Risk

To optimize above objective. We first utilize the BERT (Devlin et al., 2019) to extract intent representation. Given the i-th in-domain utterance, we get its contextual embeddings $[[CLS], T_1, T_2, ..., T_N]$. As suggested in (Zhang et al., 2021), we operate mean-pooling on these contextual token embeddings to obtain sentence semantic representation Z_i:

$$Z_i = \text{Mean-Pooling}([[\text{CLS}], T_1, ...T_N]), \quad (4)$$

where $Z_i \in \mathcal{R}^H$, N is the sequence length and H is the hidden dimension.

We optimize the empirical risk with simple softmax cross-entropy loss $\mathcal{L}_{ce}$:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\phi_{y_i}(z_i))}{\sum_{j\in[K]}\exp(\phi_j(z_i))}, \quad (5)$$

where $\phi(\cdot)$ denotes linear classifier and $\phi_j(z_i)$ denotes the score of the j-th class.

3.3 KNN-Contrastive Learning

It is worth noting that due to the lack of OOD intent samples, we can not directly optimize the open space risk. Previous methods indirectly reduce the open space risk by pulling together IND samples belonging to the same class and pushing apart samples from different classes. However, such approaches may increase the risk of identifying intra OOD intents as IND based on the above analysis. Intuitively, to reduce the risk that identifying intra OOD intents as IND, we do not need to pull together all IND intent samples belonging to the same class and just pull together k-nearest neighbors while pushing apart them from different class intent samples as shown in Figure 2 (c). In order to achieve this goal, we get the KNN-contrastive loss $\mathcal{L}_{knn-cl}$ by rewriting the contrastive loss:

$$\mathcal{L}_{\text{knn-cl}} = \sum_{i=1}^{N}\frac{1}{|\mathcal{X}_k|}\sum_{z_j\in\mathcal{X}_k} -\log\frac{\exp(\frac{z_i\cdot z_j}{\tau})}{\sum\limits_{z_q\in I}\exp(\frac{z_q\cdot z_i}{\tau})}, \quad (6)$$

where $\mathcal{X}_k$ denotes the set of k-nearest neighbors of sample z_i, $I \equiv A\bigcup\{z_j\}$, A is the set of samples whose classes are different from that of z_j. τ is the temperature hyper-parameter. We further analyze how KNN-contrastive loss is benefit to inter OOD intents and intra OOD intents simultaneously, see Appendix A.5 and Appendix A.6 for more detailed and deeper analysis.

3.4 Momentum Contrast is All You Need

When conducting KNN-contrastive learning, we need to solve two problems: a) large batch size, the more samples we can select, the more likely we to find k-nearest neighbors. Meanwhile, we also need enough negatives to distinguish. b) The k-nearest neighbors should keep consistent as they evolve during training, otherwise, KNN-contrastive train learning may be unstable. Interestingly, these problems mentioned above are also those Momentum Contrast (MoCo) (He et al., 2020) wants to solve. Following MoCo, we also maintain a queue containing IND samples and update it with features of the current batch while dequeuing the oldest features. The queue decouples the size of samples from the batch size, allowing us to obtain more negative samples (benefit to reduce open space risk). To maintain consistency, the features coming from the previous several batches are encoded by a slowly updating network (encoder), whose parameters are momentum-based average of the parameters from the query encoder (another network), see (He et al., 2020) for details. Combing softmax cross-entropy loss and KNN-contrastive learning loss, the **final finetune obejective** to learn discriminative features as:

$$\mathcal{L}_{obj} = \lambda\cdot\mathcal{L}_{\text{knn-cl}} + (1-\lambda)\cdot\mathcal{L}_{\text{ce}}, \quad (7)$$

where λ is a hyper-parameter to balance empirical and open space risk.

3.5 Local Outlier Factor

To be closer to the realistic scenario, we prefer the detection algorithm downstream without assuming a potential distribution of the IND intents. Therefore, we adopt a simple and universal detection algorithm LOF algorithm (Breunig et al., 2000) and compute LOF score following (Lin and Xu, 2019), see Appendix A.3 for specific calculation steps. Our model architecture is shown in Figure 3.

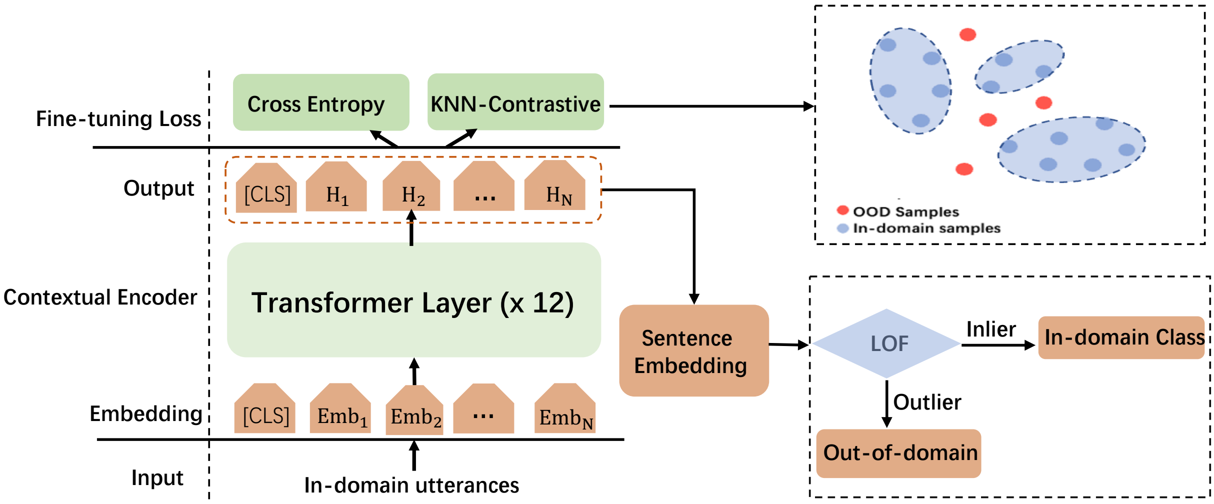


Figure 3: The architecture of our proposed model.

4 Experiments

4.1 Datasets

In order to verify the effectiveness and generality of our method, we conduct experiments on four different and challenging real-world datasets. The detailed statistics are shown in Appendix A.4.

CLINC-FULL (Larson et al., 2019) is a dataset specially designed for OOD detection, which consists of 150 classes from 10 domains. This dataset includes 22500 IND utterances and 1200 OOD utterances.

CLINC-SMALL (Larson et al., 2019) is the CLINC-FULL variant, in which there are only 50 training utterances per each IND class. This dataset includes 15000 IND utterances and 1200 OOD utterances.

BANKING (Casanueva et al., 2020) is a dataset about banking. The dataset, covering 77 classes, consists of 9003, 1000 and 3080 utterances in training, validation and test sets respectively.

StackOverflow (Xu et al., 2015) is a dataset published in Kaggle.com. This dataset has 20 classes and consists of 12000, 2000 and 6000 in training, validation and test sets respectively.

4.2 Baselines

We extensively compare our method with the following OOD classification methods: MSP (Hendrycks and Gimpel, 2017), DOC (Shu et al., 2017), SEG (Yan et al., 2020), LMCL (Lin and Xu, 2019), Softmax (Zhan et al., 2021), OpenMax (Bendale and Boult, 2016), ADB (Zhang et al., 2021), SCL (Zeng et al., 2021a).

For a fair comparison, all methods use **BERT** as the backbone network. We report the current best results of various methods on the corresponding datasets. Softmax/LMCL learns discriminative features by softmax/large margin cosine loss and use additional detector such as LOF or GDA for detecting out-of-domain. ADB (Zhang et al., 2021)/SCL (Zeng et al., 2021a) are also related to our method, however, the original paper does not report results in on some datasets. We supplement results by running their released code.

4.3 Evaluation Metrics

For all datasets, we follow previous work (Zhang et al., 2021; Zeng et al., 2021a; Zhan et al., 2021) and group all OOD classes as one rejected class. We calculate accuracy and F1-score in the same way as (Zeng et al., 2021a). To better evaluate the ability of our method to distinguish IND and OOD intents, we calculate macro F1-score over IND classes and OOD classes, represented by **F1-IND** and **F1-OOD** respectively. To comprehensively evaluate the performance of our model, we also compare accuracy score (**ACC-ALL**) and macro F1-score (**F1-ALL**) over all classes (IND and OOD classes).

4.4 Experimental Setting

Due to no OOD classes in the BANKING and StackOverflow, we follow the setting in (Zhang et al., 2021; Zhan et al., 2021). After datasets are split into train, validation, and test respectively, we randomly sample 25%, 50%, and 75% of the intent classes and discard the remaining classes in the training and validation sets [3]. The disposed classes are kept in the test set as OOD classes. CLINC-FULL and CLINC-SMALL are constructed for OOD detection especially and the datasets themselves provide OOD classes. We follow (Zeng et al., 2021a,b) and take the OOD class provided by datasets as the objective of detecting without dividing datasets additionally. As a reminder, we do not use OOD classes during training in any cases.

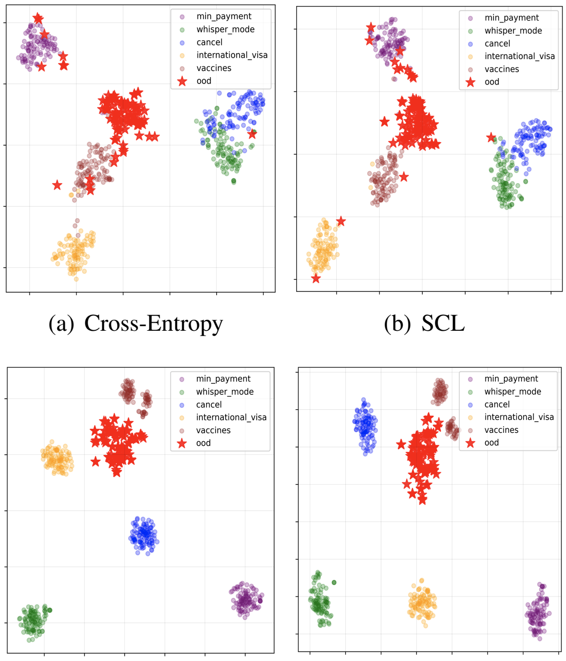
To reduce the deviation, we use two basic distances, Euclidean and Cosine (more discussed in Appendix A.2), to calculate the LOF score. For each distance, we take different random seeds to run 3 rounds, and we report the total average results. We employ the BERT model (bert-uncased, with 12-layer transformer) implemented by Huggingface Transformers[4] and adopt most of its suggested hyperparameters for finetuning. We tried learning rate in {1e-5, 5e-5}, and training batch size is set 16 or 32. Concerning contrastive learning, we tried the size of the queue in {6500, 7500, 8000} and the momentum update parameter m = 0.999. We use the AdamW optimizer (Loshchilov and Hutter, 2019). We select the LOF threshold by calculating the best macro F1-score and accuracy over IND classes on the validation set. ALL experiments were conducted in the Nvidia Ge-Force RTX-2080 Graphical Card with 11G graphical memory.

4.5 Main Results

The results in BANKING and StackOverflow are presented in Table 1, where the best results are highlighted in bold. Compared with other baselines, our method consistently improves in all settings. F1-OOD represents the F1-score of OOD class, and F1-IND is the macro F1-score of IND classes. Our method achieves favorable performance simultaneously, which shows that our method improves the capability of detecting OOD intents without sacrificing the accuracy of IND classes classification.

The results in CLINC-FULL and CLINC-SMALL datasets are presented in Table 2. Our method is also better than all other kinds of methods. It is worth noting that F1-IND and F1-OOD have improved compared with SCL significantly. We suppose this is due to the use of Momentum Contrast framework in our method, which obtains more negative samples by maintaining a queue (the capacity is much larger than batch size) so that further pushes apart samples from different classes, as shown in Section 5.1.

5 Analysis



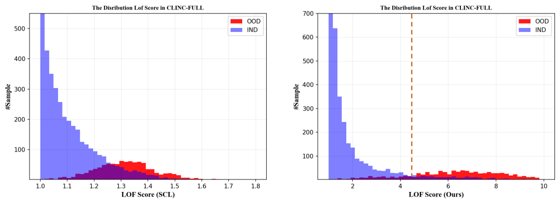
(a) Cross-Entropy

(b) SCL

(c) Ours (Top-k=5)

(d) Ours (Top-k=15)

Figure 4: t-SNE visualization of deep learned features of some intent samples in CLINC-FULL. Top-k means we select the k-nearest neighbors of IND samples for constrastive learning.



(a) SCL

(b) Ours

Figure 5: The LOF score distrubtion in CLINC-FULL. Left: SCL training. Right: Our proposed method training. The X-axis represents the LOF score, and the Y-axis represents the number of samples falling into the corresponding interval.

5.1 Feature Visualization

To compare our method with the previous methods intuitively, we use t-SNE (Van der Maaten and Hinton, 2008) to visualize deep features of some intent samples sampled from CLINC-FULL learned by BERT, SCL, and our method. Firstly, compared

[3] See more discusses in Appendix A.7.

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

	Methods	BANKING				StackOverflow			
		ACC-ALL	F1-ALL	F1-OOD	F1-IND	ACC-ALL	F1-ALL	F1-OOD	F1-IND
25%	MSP†	43.67	50.09	41.43	50.55	28.67	37.85	13.03	42.82
	DOC†	56.99	58.03	61.42	57.85	42.74	47.73	41.25	49.02
	OpenMax†	49.94	54.14	51.32	54.28	40.28	45.98	36.41	47.89
	Softmax∗	57.88	58.32	62.52	58.10	46.17	50.78	42.52	51.83
	LMCL†	64.21	61.36	70.44	60.88	47.84	52.05	49.29	52.60
	SEG∗	51.11	55.68	53.22	55.81	47.00	52.83	46.17	54.16
	ADB†	78.85	71.62	84.56	70.94	86.72	80.83	90.88	78.82
	SCL+GDA‡	83.87	67.94	89.44	66.81	82.29	70.92	88.99	67.44
	SCL+LOF‡	84.05	74.86	89.01	74.12	80.10	78.51	84.45	77.32
	Ours	**85.62**	**77.13**	**90.19**	**76.44**	**89.04**	**81.61**	**92.7**	**79.39**
50%	MSP†	59.73	71.18	41.19	71.97	52.42	63.01	23.99	66.91
	DOC†	64.81	73.12	55.14	73.59	52.53	62.84	25.44	66.58
	OpenMax†	65.31	74.24	54.33	74.76	60.35	68.18	45.00	70.49
	Softmax∗	67.44	74.19	60.28	74.56	65.96	71.94	56.80	73.45
	LMCL†	72.73	77.53	69.53	77.74	58.98	68.01	43.01	70.51
	SEG∗	68.44	76.48	60.42	76.90	68.50	74.18	60.89	75.51
	ADB†	78.86	80.90	78.44	80.96	86.40	85.83	87.34	85.68
	SCL+GDA‡	79.38	79.84	79.97	79.83	82.31	79.54	84.42	79.04
	SCL+LOF‡	80.54	82.4	80.42	82.6	84.47	84.57	85.01	84.53
	Ours	**83.14**	**83.87**	**83.58**	**83.88**	**87.62**	**87.18**	**88.36**	**87.06**
75%	MSP†	75.89	83.60	39.23	84.36	72.17	77.95	33.96	80.88
	DOC†	76.77	83.34	50.60	83.91	68.91	75.06	16.76	78.95
	OpenMax†	77.45	84.07	50.85	84.64	74.42	79.78	44.87	82.11
	Softmax∗	78.20	84.31	56.90	84.78	77.41	82.28	54.07	84.11
	LMCL†	78.52	84.31	58.54	84.75	72.33	78.28	37.59	81.00
	SEG∗	78.87	85.66	54.43	86.20	80.83	84.78	62.30	86.28
	ADB†	81.08	85.96	66.47	86.29	82.78	85.99	73.86	86.80
	SCL+GDA‡	79.86	85.14	64.49	85.5	80.88	84.79	68.83	85.86
	SCL+LOF‡	81.56	86.97	65.05	87.35	80.92	83.98	71.71	84.79
	Ours	**81.77**	**87.07**	**67.66**	**87.41**	**83.85**	**87.06**	**74.20**	**87.92**

Table 1: Results of OOD classificaion with different IND classes rate (25%, 50% and 75%) on BANKING and StackOverflow. The baseline with † are retrieved from (Zhang et al., 2021), results with ∗ are from (Zhan et al., 2021) and ‡ means the results is not provided in the original paper (Zeng et al., 2021a), and we get the results by running its released code.

Methods	CLINC-FULL				CLINC-SMALL			
	ACC-ALL	F1-ALL	F1-IND	F1-OOD	ACC-ALL	F1-ALL	F1-IND	F1-OOD
Softmax	-	-	88.98	66.17	-	-	86.20	62.58
LMCL	-	-	89.12	66.80	-	-	86.64	63.11
SCL+GDA	-	-	90.03	68.21	-	-	88.30	65.01
SCL+LOF†	84.87	88.51	88.63	70.05	84.12	87.47	87.58	70.31
Ours	**89.45**	**92.5**	**92.61**	**76.36**	**88.62**	**91.82**	**91.92**	**75.74**

Table 2: Results of OOD classification on CLINC-FULL and CLINC-SMALL. † means the results is not provided in the original paper and we get the results by running its released codes provided by (Zeng et al., 2021a). Other baselines are from (Zeng et al., 2021a).

with SCL Figure 4(b), our method further push apart samples from different classes, especially for the classes of **whisper-mode(green)** and **cancel(blue)**. This shows our method can ensure the classification quality of IND intents and better optimize empirical risk. Can our method optimize open space risk better? As shown in Figure 4(c) and Figure 4(d), we notice that the features of **vaccines class (brown)** can be clustered into three clusters (top-k=5) or two clusters (top-k=15), which means leave more space for OOD intent samples as expected.

5.2 Visualize the Ability to Detect OOD

To further verify the effectiveness of our method to characterize the difference between IND and OOD intents, we draw the LOF score distribution of samples in CLNIC-FULL test set. We show results in Figure 5. Compared with SCL (left) Figure 5(b), our LOF scores of OOD intent samples are spread in a larger range, which indicates that there is a larger margin around OOD intent samples according to the definition of LOF and further shows our method optimize the open space risk better. And this larger margin makes it is easier for us to find a baseline (brown dotted line) in Figure 5(b) (our method) to separate IND and OOD samples, which indicates they are distinguished better. The above results also ensure that we can detect OOD intents without making assumptions about the feature distribution.

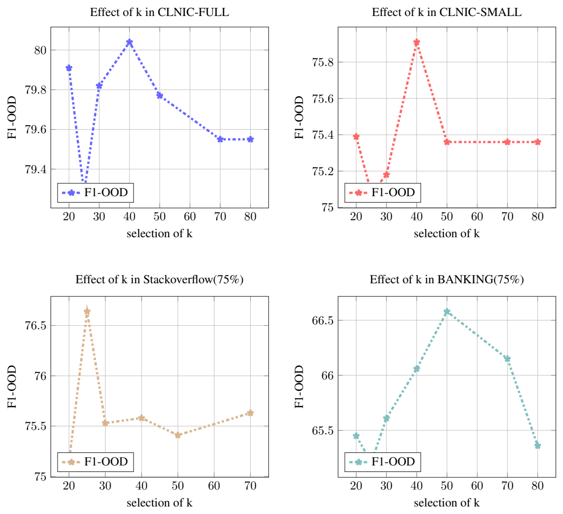


Figure 6: Effect of k. The X-axis represents the value of k, and Y-axis represents F1-score for OOD samples.

5.3 Importance of K Nearest Neighbors

In Section 3.3, we propose a novel method, which utilizes the k-nearest neighbors of IND intents to learn discriminative features, to reduce open space risk. To investigate the effect of k, we compare the performance of the model in detecting OOD intents with different k values (in a certain range) during contrastive learning (fixing other hyper-parameters). As shown in Figure 6, we have observed that the performance (cosine-based) of the model first increase and then decrease on four datasets as the value increases. This phenomenon is as expected. At the beginning of k growth, due to the reduction of open space risk, the risk of inter and intra OOD samples being identified as IND decreases (corresponding F1-OOD increases). Later, due to the compression of IND semantic space, more and more intra OOD samples are identified as IND (corresponding F1-OOD decreases) and finally tend to be stable. The phenomenon also shows that our method can better reduce the open space risk than the previous methods (which pull together all IND intents belonging to the same class).

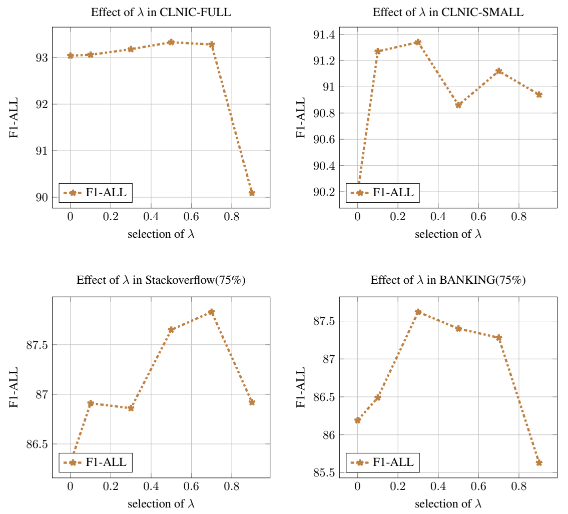


Figure 7: Effect of λ on four datasets. The X-axis represents the λ value, and Y-axis represents F1-ALL for all classes.

5.4 Trade-off Between Empirical And Open Space Risk

While we want to minimize open space risk (helpful to detect OOD intents), we also need to balance it against the empirical risk (ensure the classification quality of IND classification) over the training data. In the objective (in Eq.(7)) of OOD classification, we use the λ hyper-parameter to balance empirical and open space risk. To analyze the effect of our introduced λ, we experiment our method (cosine-based) with different λs in different datasets. As shown in Figure 7, we find with the increment of λ, the model gradually reaches the best empirical-open trade-off, which means the model can ensure the classification quality of IND and OOD detection effectively. The only empirical risk or open space risk can not make the model achieve better results.

6 Conclusion

In this paper, we explicit the optimization objective of OOD intent classification. We analyze the limitation of existing methods and propose a simple yet effective method to learn discriminative semantic features. Our approach pulls together k-nearest neighbors of IND intents and pushes apart them from different class samples to better reduce both empirical risk and open space risk. Extensive experiments conducted on four challenging datasets show our approach achieves consistent improvements without restrictions on feature distribution.

Acknowledgements

We thank Demin Song for helpful discussion about experiment. We would also like to thank Hang Yan and TianXiang Sun for helpful suggestion for this paper. This work was supported by the National Key Research and Development Program of China (No. 2020AAA0108702) and National Natural Science Foundation of China (No. 62022027).

Ethical Considerations

The datasets used in all experiments are derived from previously published scientific papers, and to our knowledge, there are no privacy or ethical issues.

References

Abhijit Bendale and Terrance E. Boult. 2015. Towards open world recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902.

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

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

DongHyun Choi, Myeongcheol Shin, EungGyun Kim, and Dong Ryeol Shin. 2021. Outflip: Generating out-of-domain samples for unknown intent detection with natural language attack. *CoRR*, abs/2105.05601.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2021. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597.

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.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 1311–1316. Association for Computational Linguistics.

Linyang Li, Demin Song, Ruotian Ma, Xipeng Qiu, and Xuanjing Huang. 2021. Knn-bert: Fine-tuning pre-trained models with knn classifier.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2021. A survey of transformers. *arXiv preprint arXiv:2106.04554*.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13675–13682. AAAI Press.

Xipeng Qiu, TianXiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *SCIENCE CHINA Technological Sciences*, 63(10):1872–1897.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2012. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Lei Shu, Hu Xu, and Bing Liu. 2017. DOC: deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2911–2916. Association for Computational Linguistics.

David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning*, 54(1):45–66.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. 2018. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9117–9126.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

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.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060.

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, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 870–878. Association for Computational Linguistics.

Zhiyuan Zeng, Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2021b. Adversarial generative distance-based classifier for robust out-of-domain detection. In *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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3521–3532. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14374–14382.

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.

A Appendix

Dataset	Classes	$Rate^*$
CLINC-FULL	150	57%
CLINC-SMALL	150	36%
StackOverflow	20	21%

Table 3: Classes denote the total number of IND classes. $Rate^*$ denotes the rate of IND classes obeying Gaussian.

A.1 Gaussian Hypothesis Testing

We use NormalTest provided by Scipy[5] to test whether the trained distribution conforms to a normal distribution. The Multivariate Normal Testing is not adopted mainly because samples with 256 dimensions can hardly fit the multivariate Gaussian distribution. Moreover, the number of samples will affect the p-value considerably. So we randomly select 50 samples from each class and calculate the p-value on individual dimensions separately. To mitigate the influence of some too small values, we retain the highest 128 value for statistics. By convention, the distribution with a p-value less than 0.05 will reject the Normal Hypothesis. We calculate the percentage of not rejected dimensions and report the mean value of all classes. We conduct Gaussian hypothesis testing on the IND semantic features distribution, which are learned by (Zeng et al., 2021a), in the train sets of CLINC-FULL, CLINC-SMALL, and StackOverflow. Results are shown in Table 3.

	BANKING(75%)		StackOverflow(75%)	
	F1-OOD	F1-ALL	F1-OOD	F1-ALL
Lof Cosine	67.36	86.87	78.12	88.06
Lof Euclidean	65.77	87.46	77.12	87.9

Table 4: Results of OOD classification on BANKING and StackOverflow. Lof Cosine means to calculate Lof based on Cosine and Lof Euclidean means to calculate Lof based on Euclidean.

A.2 Effect of Distance on Calculation of LOF

We calculate LOF-score based on two distances: Cosine and Euclidean. In this section, we want to investigate the effect of LOF-score calculated by different distances. On the two datasets, we use these two distances to calculate LOF score (average results over 3 runs randomly) and get F1 for out-of-domain and macro F1 for all classes. As shown in Table 4, Firstly, from a single dataset, there are some differences between the two methods (although the difference is not very large). We guess this may be caused by the threshold. Because we can't use the information of OOD samples, we can't accurately obtain the threshold required by each method. Secondly, comparing the performance of the same method on different datasets, we have not observed one method has more advantages. Therefore, to comprehensively illustrate the effectiveness of our method, in the experiment we take different random seeds to test 3 times for each distance, and we report the total average results. It is worth noting that during our experiment, we find that raising the threshold for LOF-score calculated based on Euclidean (compared with Cosine) can achieve better average results for CLINC-FULL and CLINC-SMALL datasets.

A.3 Local Outlier Factor

To be closer to the realistic scenario, we prefer detection in downstream without relying on assumption of distribution. Therefore, we adopt a simple and universal detection algorithm LOF algorithm (Breunig et al., 2000). we compute LOF score follow (Lin and Xu, 2019):

$$\mathrm{LOF}_k(z) = \frac{\sum_{p \in N_k(z)} \frac{lrd(p)}{lrd(z)}}{|N_k(z)|}, \tag{8}$$

where $N_k(z)$ denotes the set of k-nearest neighbors of z. The LOF captures the degree to which z is

[5] https://github.com/scipy/scipy

Dataset	Class	#Training	#Validation	#Test	Vocabulary Size	Length (Avg)
CLINC-FULL	150	15100	3100	5500	8288	8.32
CLINC-SMALL	150	7600	3100	5500	7017	8.31
BANKING	77	9003	1000	3080	5028	11.91
StackOverflow	20	12000	2000	6000	17182	9.18

Table 5: Statistics of datasets. # denotes the total number of utterances

an outlier by computing the average of the ratio of the local reachability density of z and the k-nearest neighbors of z.

Local Reachability Density (lrd) computed as following:

$$lrd_k(p) = 1/\frac{\sum_{O\in N_k(p)} \text{reach-dist}_k(p,o)}{|N_k(p)|}, \quad (9)$$

where $lrd_k(p)$ denotes the inverse of the average reachability distance based on the k-nearest neighbors of p. reach-dist$_k$(p, o) is defines as:

$$\text{reach-dist}_k(p,o) = \max\{\text{k-dis}(o), d(p,o)\}, \quad (10)$$

where d(p, o) is the distance between p and o, and k-dis is the distance of o to its k-th nearest neighbor. It is easy to see that the lower lrd of a sample is than its neighbors, the more likely it is to be an OOD sample.

A.4 Statistics of Datasets

The datailed statistics of **CLINC-FULL**, **CLINC-SMALL**, **BANKING** and **StackOverflow**, which are used in Section 4.1, are shown in Table 5.

A.5 Theoretical Analysis

In this section, we take a closer look at how our proposed KNN-contrastive loss indirectly can reduce the risk that inter OOD and intra OOD intents are identified as IND.

$$\mathcal{L}_{\text{knn-cl}} = \sum_{i=1}^{N} \frac{1}{|\mathcal{X}_k|} \sum_{z_j\in\mathcal{X}_k} \underbrace{-\log \frac{\exp(\frac{z_i\cdot z_j}{\tau})}{\sum_{z_k\in I}\exp(\frac{z_k\cdot z_i}{\tau})}}_{\mathcal{R}} . \quad (11)$$

For simplicity, we consider KNN-contrastive loss with one sample q, denoted as $\mathcal{R}$ in Eq. (11):

$$\mathcal{R} = \sum_{z_j\in\mathcal{X}_k} -log\frac{\exp(\frac{q\cdot z_j}{\tau})}{\sum\limits_{z_k\in I} exp(\frac{q\cdot z_k}{\tau})}, \quad (12)$$

where $\mathcal{X}_k$, I have the same meanings as in Eq.(11), representing the set of k-nearest neighbors of q, $I \equiv A\bigcup\{z_j\}$, A is the set of samples whose classes are different from that of z_j. We introduce a term on both numerator and denominator, $\sum_{z_p\in I^+} exp(\frac{q\cdot z_p}{\tau})$, I^+ denotes set of all samples whose classes are the same as q, so $\mathcal{R}$ can be rewritten as:

$$\begin{aligned}\mathcal{R} &= \sum_{z_j\in\mathcal{X}_k} -log\frac{\exp(\frac{q\cdot z_j}{\tau})}{\sum\limits_{z_k\in I} exp(\frac{q\cdot z_k}{\tau})} \\ &= \sum_{z_j\in\mathcal{X}_k} -log\frac{\exp(\frac{q\cdot z_j}{\tau})}{\sum\limits_{z_p\in I^+}\exp(\frac{q\cdot z_p}{\tau})}\frac{\sum\limits_{z_p\in I^+(i)}\exp(\frac{q\cdot z_p}{\tau})}{\sum\limits_{z_k\in I} exp(\frac{z_k\cdot q}{\tau}))} \\ &= \sum_{z_j\in\mathcal{X}_k} \underbrace{-log\frac{\exp(\frac{q\cdot z_j}{\tau})}{\sum\limits_{z_p\in I^+}\exp(\frac{q\cdot z_p}{\tau})}}_{\mathcal{R}_{\mathcal{O}}^{intra}(f)} \underbrace{-log\frac{\sum\limits_{z_p\in I^+}\exp(\frac{q\cdot z_p}{\tau})}{\sum\limits_{z_k\in I} exp(\frac{z_k\cdot q}{\tau})}}_{\mathcal{R}_{\mathcal{O}}^{inter}(f)} .\end{aligned} \quad (13)$$

In this way, the KNN-contrastive loss can be split into two comparative learning losses. The first loss $\mathcal{R}_{\mathcal{O}}^{intra}(f)$ is to pull together k-nearest neighbors of q and push apart q from other samples whose classes are the same as q, which is viewed to reduce the risk identifying intra OOD samples as IND. The second loss $\mathcal{R}_{\mathcal{O}}^{inter}(f)$ is to push together samples whose classes are the same as q and pull apart samples whose classes are different q, which is viewed to reduce the risk identifying inter OOD samples as IND.

A.6 Specific Open Space Risk

Intra-space risk and Inter-space risk When we encounter a scene with limited OOD samples, we can re-explore the nature of open space risk. Different from (Scheirer et al., 2012; Bendale and Boult, 2015), we define open space as the space around OOD samples:

$$\mathcal{O} = \bigcup_{x\in\phi}\sigma(x) = \underbrace{\bigcup_{x\in\phi_{in}}\sigma(x)}_{\mathcal{O}_{in}} + \underbrace{\bigcup_{x\in\bar{\phi}_{in}}\sigma(x)}_{\bar{\mathcal{O}}_{in}}, \quad (14)$$

where σ is a local and small semantic space spanned by OOD samples, ϕ is the set of all OOD samples (or just included enough OOD samples), ϕ_{in} is set of OOD samples within IND distribution (surrounded by a convex hull composed of IND samples locally), $\bar{\phi}_{in}$ is set of OOD samples between IND classes ($\bar{\phi}_{in} = \phi - \phi_{in}$). Consider a large space $\mathcal{S}$ including both open space $\mathcal{O}$ and remaining space, and a measurable recognizer(or discriminator) f that $f(x) = 1(> 0)$ for recognition of the IND samples, otherwise $f(x) = 0(<= 0)$. Probabilistic (in terms of Lebesgue measure) open space risk $\mathcal{R}$ can also be formalized as:

$$\mathcal{R}_{\mathcal{O}}(f) = \frac{\int_{\mathcal{O}} f(x,\theta)dx}{\int_{\mathcal{S}} f(x,\theta)dx}. \qquad (15)$$

Considering Eq.(14), we can naturally decompose open space risk into two terms shown in Eq.(16), called intra-space risk and inter-space risk respectively.

$$\mathcal{R}_{\mathcal{O}}(f) = \underbrace{\frac{\int_{\mathcal{O}_{in}} f(x,\theta)dx}{\int_{\mathcal{S}} f(x,\theta)dx}}_{\mathcal{R}_{\mathcal{O}}^{intra}(f)} + \underbrace{\frac{\int_{\bar{\mathcal{O}}_{in}} f(x,\theta)dx}{\int_{\mathcal{S}} f(x,\theta)dx}}_{\mathcal{R}_{\mathcal{O}}^{inter}(f)}. \qquad (16)$$

By aligning Eq.(16) with Eq.(13), we can naturally find out why KNN-contrastive loss can better reduce open space risk.

	Methods	BANKING			
		ACC-A	F1-A	F1-O	F1-I
25%	ADB†	78.85	71.62	84.56	70.94
	Ours	**91.26**	**84.1**	**93.19**	**83.57**
50%	ADB†	78.86	80.90	78.44	80.96
	Ours	**82.51**	**82.64**	**82.96**	**82.62**
75%	ADB†	81.08	85.96	66.47	86.29
	Ours	**81.62**	**87.17**	**66.56**	**87.52**

Table 6: Results of OOD classificaion with different IND classes rate (25%, 50% and 75%) on BANKING. The baseline with † are retrieved from (Zhang et al., 2021).

A.7 Random Split Matters?

In the experiment, we noticed a small and interesting problem. When conducting experiments on BANKING and StackOverflow, there are two evaluation methods. The first is as described in the experimental section Section 4.4. Another evaluation method is that the dataset is fixed after the known intents classes are randomly sampled from the datasets, and the random test is carried out on these fixed datasets. We also take different random seeds to run 3 rounds and compare our results with ADB. The results are presented in Table 6 and Table 7. Our results are also better than ADB on all datasets. More interestingly, the results even can be better than our results in Section 4.4 on some datasets (25% Banking). We speculate that this may be because this evaluation method may achieve a better split in some cases for learning feature representation. In addition to in the Nvidia GeForce RTX-2080 Graphical Card with 11G graphical memory, we also conducted some experiments in the Nvidia Ge-Force RTX-3090 Graphical Card with 24G graphical memory.

	Methods	StackOverflow			
		ACC-A	F1-A	F1-O	F1-I
25%	ADB†	86.72	80.83	90.88	78.82
	Ours	**89.28**	**82.97**	**92.85**	**81.00**
50%	ADB†	86.40	85.83	87.34	85.68
	Ours	**87.87**	**87.28**	**88.79**	**87.15**
75%	ADB†	82.78	85.99	73.86	86.80
	Ours	**84.98**	**87.49**	**76.95**	**88.20**

Table 7: Results of OOD classificaion with different IND classes rate (25%, 50% and 75%) on StackOverflow. The baseline with † are retrieved from (Zhang et al., 2021).