# A Closer Look at Few-Shot Out-of-Distribution Intent Detection

**Li-Ming Zhan**[1] **Haowen Liang**[1*] **Lu Fan**[1*] **Xiao-Ming Wu**[1†]
**Albert Y.S. Lam**[2]
Department of Computing, The Hong Kong Polytechnic University, Hong Kong S.A.R.[1]
Fano Labs, Hong Kong S.A.R.[2]
{lmzhan.zhan, michael.liang}@connect.polyu.hk
{cslfan, csxmwu}@comp.polyu.edu.hk, albert@fano.ai

## Abstract

We consider few-shot out-of-distribution (OOD) intent detection, a practical and important problem for the development of task-oriented dialogue systems. Despite its importance, this problem is seldom studied in the literature, let alone examined in a systematic way. In this work, we take a closer look at this problem and identify key issues for research. In our pilot study, we reveal the reason why existing OOD intent detection methods are not adequate in dealing with this problem. Based on the observation, we propose a promising approach to tackle this problem based on latent representation generation and self-supervision. Comprehensive experiments on three real-world intent detection benchmark datasets demonstrate the high effectiveness of our proposed approach and its great potential in improving state-of-the-art methods for few-shot OOD intent detection. The source code can be found at https://github.com/liam0949/Few-shot-Intent-OOD.

## 1 Introduction

Intent detection is an important component of task-oriented dialogue system, which aims at accurately identifying the intent behind user utterances. Out-of-distribution (OOD) intent detection aims to solve a $(K+1)$-way classification problem with $K$ in-distribution (ID) intent classes and an additional OOD class representing malformed or unsupported queries. In practice, OOD intent detection is often performed in data-scarcity scenarios, e.g., at the early development stage of a dialogue system when labeled data is not sufficient, or for dialogue systems developed for minority language users where it is difficult to find suitable annotators.

Despite its practical importance, few-shot OOD intent detection is a highly challenging problem, which is seldom studied in the literature and has not been investigated in a systematic way. Recent advances in OOD intent detection (Zhang et al., 2021a; Zhan et al., 2021; Lin and Xu, 2019) commonly assume that there are adequate ID examples available for training, without considering the few-shot scenario. To our best knowledge, the only work on this topic is by Zhang et al. (2020), who try to tackle few-shot OOD intent detection via transfer learning by fine-tuning RoBERTa (Liu et al., 2019) on large-scale natural language inference datasets.

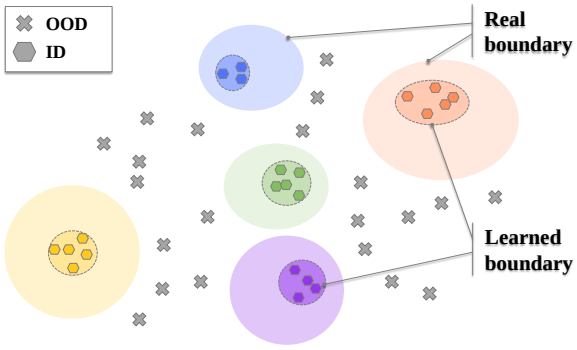


Figure 1: The challenge of few-shot out-of-distribution intent detection. OOD stands for out-of-distribution examples and ID stands for in-distribution examples.

In this work, we take a closer look at few-shot OOD intent detection and consider a strict setting, where only few-shot in-distribution labeled examples are available during training and no external resources can be exploited, since the requirement of additional resources hinders the applicability of the model. Under this simplified yet more challenging setting, state-of-the-art OOD intent detection algorithms fail to achieve acceptable performance. In Figure 1, we illustrate the key challenge for few-shot OOD detection. As shown in Figure 1, since ID classes are under-represented by few-shot ID examples, a model based on density estimation (Zhang et al., 2021a) or $(K+1)$-way discriminative training (Zhan et al., 2021) tends to learn a conservative decision boundary and hence there

*Equal contribution.
† Corresponding author.



*Proceedings of the 29th International Conference on Computational Linguistics*, pages 451–460
October 12–17, 2022.

are large margins between the real and learned decision boundaries. Real ID examples situate in the margins will be inaccurately assigned to the OOD class, leading to poor performance.

Therefore, the key for few-shot OOD intent detection is to improve the model performance on ID examples. To address this issue, we propose to enrich the training set to improve the representativeness of ID intent classes and provide more useful learning signals. We explore the feasibility of generating synthetic ID examples in a self-supervised manner. In particular, we train a denoising autoencoder (DAE) (Vincent et al., 2008) in the latent representation space only using the few labeled ID examples. The trained decoder of DAE is then used to efficiently sample synthetic ID examples. With the enlarged training set, we follow Zhan et al. (2021) to train a $(K+1)$-way classifier by simulating OOD examples with the enlarged training set. Our contributions are summarized as follows:

- We pioneer in studying a practical but more challenging few-shot OOD intent detection problem and identifying the key challenge for this problem.
- We propose a promising approach for solving few-shot OOD intent detection based on latent representation generation and $(K+1)$-way discriminative training, which requires no additional resources for training and validation.
- We conduct comprehensive experiments on three realistic intent detection datasets to verify the effectiveness and robustness of our method in diverse few-shot OOD intent detection scenarios.

## 2 Related Work

**Out-of-Distribution Intent Detection.** Out-of-distribution (OOD) intent detection (or out-of-domain intent detection) has attracted much attention in research communities, due to its significant importance to the robustness of dialogue systems. The primary challenge of this task is that there is no labeled OOD example available for training and validation. As such, the majority of OOD intent detection algorithms relies on manually selecting an appropriate threshold.

The first line of works (Hendrycks and Gimpel, 2017; Shu et al., 2017; Ryu et al., 2018, 2017) uses some statistic as the confidence score of whether an example is OOD or not. Hendrycks and Gimpel (2017) pointed out that the negative probability outputted by the softmax function can be a good confidence metric for OOD detection. Shu et al. (2017) defined a binary classification task for every in-domain class and used the maximum probability among all these binary classifiers as the confidence score. Ryu et al. (2018) developed an adversarial training strategy inspired by GAN for OOD intent detection. The discriminator in GAN was trained to assign lower scores to OOD examples. Ryu et al. (2017) employed an autoencoder trained on in-domain examples and used the reconstruction score as the OOD indicator. However, all these methods require manual effort in selecting a proper threshold for OOD discrimination.

The second line of works (Lin and Xu, 2019; Zhang et al., 2021a; Yan et al., 2020) proposes to learn decision boundaries for OOD examples under some assumption of data distribtuion, e.g., mixture of Gaussians. OOD examples are assumed to lie in the low-density areas of utterance distribution. Yan et al. (2020) proposed to model the in-domain examples by a mixture of Gaussians distribution and select a margin to constrain the variance of each in-domain Gaussian component. Zhang et al. (2021a) also made the mixture of Gaussian assumption on in-domain data distribution but proposed to automatically learn the variance of the Gaussian components.

Different from previous methods, a recent work by Zhan et al. (2021) proposed to directly learn a $(K+1)$-way classifier in an end-to-end manner. They created OOD learning signals during training by leveraging external data or constructing simulated OOD examples with self-supervised information.

**Few-shot OOD Intent Detection.** Few-shot OOD intent detection considers OOD intent detection in low-resource scenarios. It aims at developing a reliable OOD detector with only a few examples per each in-distribution class. Undoubtedly, this is a highly challenging task given that few-shot intent detection is already a big challenge (Zhang et al., 2020). At this point, this task is under-explored and has never been investigated in a strictly low-resourced setting. The most related work is DNNC proposed in Zhang et al. (2020), which tries to mitigate the data-scarcity problem by fine-tuning RoBERTa on external large natural language inference datasets. In this paper, however,

we consider using the few-shot labeled examples as the only training resource.

**General-purpose Few-shot OOD Detection.** There is also little research on general-purpose few-shot OOD detection. To our knowledge, recent works are Jeong and Kim (2020) and Wang et al., both of which adopt episodic training on a large set of few-shot classification tasks for transfer learning. Clearly, this is very different from the problem setting of this paper, as we do not use training resources other than the given few-shot labeled examples.

## 3 Problem Statement and Pilot Study

**Out-of-distribution (OOD) intent detection** aims at improving the robustness of a dialogue system with respect to utterances with unknown (or unsupported) intents. The key challenge of OOD detection is that real OOD samples are inaccessible during training and validation. Given an in-distribution (ID) set of $K$ known classes, $y_i \in \{y^k\}_{k=1}^K$, the OOD detection task considers another special OOD class $y^{OOD}$ to represent any malformed or unsupported utterances. Hence, given the input space $\mathcal{X} \times \mathcal{Y}$, the goal of OOD intent detection is to learn a $(K+1)$-way classifier $f_\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ to minimize the expected risk:

$$R(f) = \mathbb{E}(\mathbb{1}[f_\phi(x_i) \neq y_i)]), \quad (1)$$

where $y_i \in \{y^1, \cdots, y^K, y^{\text{OOD}}\}$ and the expectation is taken over the joint distribution of $p(x, y)$. $\mathbb{1}$ is an indicator function.

**Few-shot OOD intent detection** is a more challenging setting with the assumption that there are only a few labeled in-distribution (ID) examples available during training. In this paper, we consider a strict but practical setting by assuming that there are no additional resources (e.g., labeled or unlabeled auxiliary datasets) available to aid the training of the classifier $f_\phi(\cdot)$ or during fine-tuning pre-trained language models. Typically, for each ID class in $\{y^k\}_{k=1}^K$, there are only $\sim 5$ or $\sim 10$ labeled examples per class.

**Pilot study.** To illustrate the challenges of few-shot OOD intent detection, we conduct a pilot study on a commonly used OOD intent detection dataset CLINC150 (Larson et al., 2019) using two recent state-of-the-art approaches (Zhang et al., 2021a; Zhan et al., 2021) for few-shot OOD intent detection. To simulate the few-shot scenario, in the experiment, only 5 labeled examples in each ID class are used for training. The results are summarized in Table 1. For OOD detection, we randomly select 25%, 50% and 75% intent classes as in-domain classes and assign the remaining classes to the OOD category. Experimental details are elaborated in Section 5.

| | Methods | Acc. | Macro-F1 | ID-F1 | OOD-F1 |
|---|---|---|---|---|---|
| 25% | ADB | 77.91 | 53.09 | 52.22 | 86.29 |
| | DCL | 86.53 | 48.78 | 47.63 | 92.22 |
| 50% | ADB | 69.36 | 56.91 | 56.64 | 77.17 |
| | DCL | 74.60 | 50.58 | 50.15 | 82.45 |
| 75% | ADB | 70.43 | 67.17 | 67.12 | 73.09 |
| | DCL | 65.50 | 54.25 | 54.11 | 70.22 |

Table 1: A pilot study on few-shot OOD intent detection. DCL (Zhan et al., 2021) and ADB (Zhang et al., 2021a) are two recent state-of-the-art approaches for OOD intent detection. ID-F1 indicates macro f1-score on the in-distribution classes. OOD-F1 stands for f1-score on the out-of-distribution class.

We can observe that both of the two methods yield unsatisfactory performance. Specifically, the performance on the ID classes is poor and way lower than that on the OOD class. When there are only 25% ID classes ($\sim 38$), the gap between the ID and OOD classes in f1-score is the largest (up to 44+). Although moderate overall accuracy is achieved, such OOD intent detection model can only provide services to users worse than random choices, since the majority of user utterances are rejected as OOD inputs. It also indicates that the overall accuracy may not be a good performance measure for this task. These observations show that in the few-shot scenario, existing OOD intent detection algorithms can be easily biased towards the OOD class, due to inadequate representations of the ID classes. Hence, directly applying them to few-shot OOD intent detection will lead to sub par performance.

The primary challenge identified from this pilot experiment for few-shot OOD intent detection is then how to improve the performance on in-distribution classes and achieve a good balance in performance between ID and OOD classes.

## 4 Methodology

### 4.1 Utterance Representation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be the training set, where $x_i$ denotes an input token sequence with size $m$, i.e., $[x_i^0, \cdots, x_i^{m-1}]$. For each input $x_i$, we use BERT as the encoder to map $x_i$ into a sequence

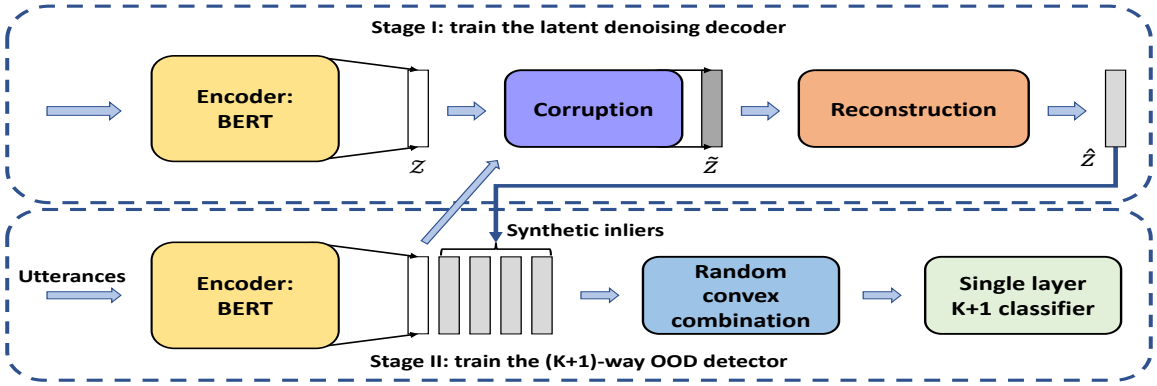


Figure 2: An overview of our proposed framework.

of hidden states $h_i$, i.e., BERT:$\mathcal{X} \rightarrow \mathcal{H}$ and $h_i \in \mathbb{R}^{(m+1)*768}$. Note that for every sentence, BERT adds a spacial token [CLS] at the beginning of the sequence. Following common practice, we use the average pooling of the hidden sequence $h_i$ as the representation of an utterance:

$$z_i = \text{Avg.Pool}([h_i^{\text{CLS}}, h_i^0, \cdots, h_i^{m-1}]).$$

Then, we obtain a mapped training set $\mathcal{D}^{tr} = \{(z_i, y_i)\}_{i=1}^N$. We instantiate few-shot OOD detector $f_\phi(\cdot)$ by replacing the pre-trained heads of BERT with a simple linear mapping layer.

### 4.2 Our Proposed Model

As shown in Figure 2, we propose a two-stage model for few-shot OOD intent detection. In the first stage, we learn a stochastic reconstruction function to generate synthetic ID samples in the representation space to enrich the in-distribution training set. In the second stage, we adopt a $(K+1)$-way discriminative training procedure for OOD detection by simulating OOD examples based on the enlarged in-distribution training set. Notice that throughout the two stages, we only use the few labeled in-distribution data without exploiting external labeled intent detection data or fine-tuning corpus.

#### 4.2.1 Stage I: Generating Synthetic In-distribution Data

To improve the performance of in-distribution (ID) classes, our solution is to learn a latent denoising autoencoder (DAE) (Vincent et al., 2008) in the latent representation space $\mathcal{Z}$ of BERT, to enrich the in-domain training set by generating synthetic examples with the reconstructor of the DAE.

Our key idea is to learn an approximator for the distribution of the latent representation of ID utterances ($p(z)$), from which we can sample synthetic ID examples. We aim to learn a generator with sampling efficiency and guaranteed consistency in approximating the true distribution as the training size $N \rightarrow \infty$. We can thereby enrich the ID training examples directly in the representation space $\mathcal{Z}$ and save the effort of conducting data augmentation in the input space $\mathcal{X}$.

To this end, we employ a principled distribution estimation method – denoising autoencoder (DAE) – to build an efficient stochastic process for sampling ID examples with a consistency guaranteed estimator for $p(z)$. The latent DAE consists of two components: the corruption distribution $\mathcal{C}(\tilde{z} \mid z)$ and the reconstruction distribution $q_\theta(z \mid \tilde{z})$. The DAE can be learned by:

$$\theta^* = \underset{\theta}{\text{argmax}}\, \mathbb{E}(\log(q_\theta(z \mid \tilde{z})),$$

where the likelihood is computed by a mean square loss between the original embedding vector $z$ and the reconstructed vector $\hat{z}$ as shown in Figure 2.

After obtaining the reconstruction distribution $q_{\theta^*}(z \mid \tilde{z})$, we can sample synthetic ID examples as follows:

$$\begin{aligned} \hat{z} &\sim q_{\theta^*}(z \mid \tilde{z}), \\ \tilde{z} &\sim \mathcal{C}(\tilde{z} \mid z). \end{aligned} \tag{2}$$

The corruption distribution $\mathcal{C}$ can be instantiated by simple stochastic operations like Dropout (Srivastava et al., 2014). By repeatedly applying the process in Equation (2), we can obtain a synthetic labeled ID set $\mathcal{D}^{\text{rec}} = \{(\hat{z}_i, y_i)\}_{i=1}^L$, where the reconstructed representation $\hat{z}_i$ shares the same label $y_i$ with the original uncorrupted $z_i$. Finally, by combining the original training set $\mathcal{D}^{tr}$ and the synthetic set $\mathcal{D}^{\text{rec}}$, we get an enlarged labeled training set $\mathcal{D}^{\text{Enlarged}} = \mathcal{D}^{tr} \cup \mathcal{D}^{\text{rec}}$.

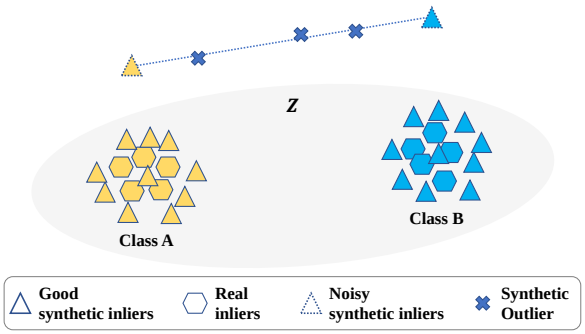


Figure 3: Illustration of the noise neutralizing effect under the $(k+1)$-way training paradigm.

#### 4.2.2 Stage II: $(K+1)$-way Discriminative Training

As shown in the Figure 2, the second stage of our proposed method aims at learning a $(K+1)$-way classifier in an end-to-end manner. Since only few-shot samples are used to train the reconstruction distribution $q_\theta(z \mid \tilde{z})$, the resulting $q_{\theta^*}(\cdot)$ may not be a perfect estimator for the true distribution, and the enlarged in-distribution set $\mathcal{D}^{\text{Enlarged}}$ may be noisy. Hence, it may not be the best choice to directly apply density estimation-based methods for OOD intent detection, due to the risk of overfitting.

To better utilize the enlarged in-distribution set $\mathcal{D}^{\text{Enlarged}}$, we adopt the $(K+1)$-way discriminative training strategy proposed in Zhan et al. (2021) and follow their idea to construct OOD learning signals via random convex combination between representations from different in-distribution classes in the enlarged in-distribution set. By doing so, the impact of noisy synthetic in-distribution examples can be mitigated. We demonstrate this phenomenon in Figure 3. The linear interpolation between off-manifold noisy synthetic in-distribution examples tends to represent the OOD examples, since the word embeddings of BERT has been found concentrating near a low-dimensional manifold of the representation space (Ethayarajh, 2019).

Specifically, given the enlarged training set $\mathcal{D}^{\text{Enlarged}}$, we construct an OOD set $\mathcal{D}^{\text{OOD}}$ by:

$$z_i^{OOD} = \alpha * z_i + (1-\alpha) * z_j, \qquad (3)$$

where $y^i \neq y^j$, $\alpha \in [0,1]$ is randomly sampled from $U(0,1)$ and $z_i,\ z_j \in \mathcal{D}^{\text{Enlarged}}$.

Finally, our $(K+1)$-way classifier can be learned by minmizing the loss in Equation (1) on the union set $\mathcal{D}^{\text{OOD}} \cup \mathcal{D}^{\text{Enlarged}}$.

## 5 Experiments

To evaluate our proposed method for few-shot out-of-distribution (OOD) intent detection, we conduct extensive experiments on three real-world benchmark datasets. By comparing with state-of-the-art OOD intent detection methods, we find that our method can outperform these baselines by a large margin, especially in extreme few-shot scenarios. Moreover, our approach yields a more consistent performance at different few-shot OOD settings, demonstrating the robustness of our algorithm.

### 5.1 Datasets and Baselines

We evaluate our method on three commonly used OOD intent detection datasets, which are introduced as follows.

- **CLINC150** (Larson et al., 2019) is specifically designed for OOD intent detection. It consists of 150 in-distribution classes with 15,000 samples for training, 3,000 for validation, and 4,500 for testing. Besides, it also contains 1,200 annotated OOD instances, and we put all the OOD examples into the test set.

- **Banking** (Casanueva et al., 2020) contains data from the banking domain, with 13,083 samples of 77 intents. We split the dataset into 9,003 for training, 1,000 for validation, and 3,080 for testing.

- **StackOverflow** (Xu et al., 2015) contains data in 20 classes, each of which contains 1,000 samples. We use 12,000 samples for training, 2,000 for validation, and 6,000 for testing.

The dataset statistics are summarised in Table 2.

To evaluate the effectiveness of our proposed method, we compare it with the following baselines.

- **MSP** (Hendrycks and Gimpel, 2017): It leverages the probabilities outputted by the softmax function for out-of-domain detection. As correct samples tend to have higher probability scores, samples below a threshold are classified as outliers. We set the threshold as 0.5 in our experiment.

- **DOC** (Shu et al., 2017): It shares a similar idea with MSP in assuming that in-distribution examples tend to have higher probability scores. It uses the maximum probability

| Dataset (proportion) | # Vocab | Avg. Length | # Training | # Class | Avg. Sample per Class (5%) | (10%) |
|---|---|---|---|---|---|---|
| CLINC150 | 5864 | 8.34 | 15000 | 150 | 5 | 10 |
| Banking | 4327 | 11.99 | 9003 | 77 | 6 | 12 |
| StackOverflow | 16519 | 8.35 | 12000 | 20 | 30 | 60 |

Table 2: Dataset statistics.

| dataset | CLINC150 | | | Banking | | | StackOverflow | | |
|---|---|---|---|---|---|---|---|---|---|
| p=5% | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 |
| MSP | 40.13 | 55.17 | 54.76 | 17.74 | 29.31 | 31.99 | 52.30 | 42.92 | 78.92 |
| DOC | 11.05 | 8.62 | 44.37 | 15.79 | 25.61 | 20.98 | 65.54 | 44.4 | 58.54 |
| SEG | 36.09 | 51.90 | 62.64 | 39.53 | 52.27 | 58.80 | 60.76 | 75.93 | 83.22 |
| LMCL | 34.30 | 52.45 | 60.71 | 39.10 | 48.90 | 54.60 | 56.00 | 69.68 | 83.17 |
| Softmax | 33.98 | 52.48 | 62.11 | 32.77 | 43.74 | 52.84 | 54.21 | 71.27 | 81.55 |
| ADB | 53.09 | 56.91 | 65.65 | 37.74 | 45.91 | 55.26 | 60.31 | 77.92 | 81.14 |
| DCL | 48.78 | 50.58 | 54.25 | 33.92 | 39.10 | 45.59 | 78.98 | 82.37 | 83.01 |
| **Ours** | **62.19** | **64.79** | **68.30** | **48.23** | **58.92** | **63.14** | **80.48** | **84.04** | **84.25** |
| dataset | CLINC150 | | | Banking | | | StackOverflow | | |
| p=10% | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 |
| MSP | 54.34 | 71.56 | 77.31 | 43.50 | 48.62 | 68.34 | 41.66 | 59.73 | 75.95 |
| DOC | 15.15 | 23.28 | 54.69 | 13.99 | 21.50 | 25.13 | 44.77 | 61.22 | 61.19 |
| SEG | 68.29 | 77.59 | 80.32 | 56.75 | 58.70 | 71.32 | 58.77 | 78.64 | 83.85 |
| LMCL | 66.87 | 76.48 | 79.04 | 54.38 | 63.71 | 67.66 | 55.42 | 77.01 | 85.06 |
| Softmax | 65.07 | 77.08 | 79.68 | 53.27 | 60.20 | 68.94 | 57.86 | 77.30 | 83.47 |
| ADB | 68.05 | 74.96 | 77.75 | 51.12 | 66.16 | 70.50 | 69.55 | 81.30 | 83.83 |
| DCL | 68.65 | 72.74 | 70.81 | 55.74 | 61.10 | 65.77 | 78.61 | 82.46 | 83.80 |
| **Ours** | **72.43** | **78.15** | **82.17** | **60.99** | **67.89** | **73.79** | **81.07** | **83.99** | **85.11** |

Table 3: Overall macro f1-score including the OOD class for few-shot OOD intent detection with different proportion (0.25, 0.5 and 0.75) of in-distribution classes. $p$ indicates the ratio of selected few-shot in-distribution examples. For each setting, the best result is marked in bold.

from $m$ 1-vs-rest sigmoid classifiers for $m$ ID classes respectively as the confidence score.

- **LMCL** (Lin and Xu, 2019): It leverages local outlier factor(LOF) to identity samples which are far away from the clusters in the embedding space as outliers. The model learns discriminative features by largin margin cosine loss.

- **Softmax** (Lin and Xu, 2019): It is a variant of LMCL where the large margin cosine loss is replaced by the softmax loss to learn discriminative features.

- **SEG** (Yan et al., 2020): It uses a Gaussian mixture model to enforce ID embeddings to form ball-like dense clusters in the feature space. Moreover, it injects semantic information into the Gaussian mixture model by assigning the embeddings of class labels or descriptions to be the means of the Gaussians.

- **ADB** (Zhang et al., 2021a): It proposes to learn a decision boundary for each in-domain class for OOD intent detection. Samples reside outside of the boundaries are identified as outliers, while in-distribution examples are classified based on their distance to centroids of each class.

- **DCL** (Zhan et al., 2021): It treats outliers as an additional class and proposes a $K+1$ training paradigm for OOD intent detection. Samples in the outlier class are obtained from external datasets and synthesized through convex combinations of in-distribution features.

| dataset | CLINC150 | | | Banking | | | StackOverflow | | |
|---|---|---|---|---|---|---|---|---|---|
| p=5% | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 |
| MSP | 38.85 | 54.85 | 54.63 | 14.38 | 28.32 | 31.79 | 55.15 | 40.97 | 80.44 |
| DOC | 8.99 | 7.72 | 44.18 | 12.35 | 24.48 | 20.62 | 62.89 | 42.89 | 58.92 |
| SEG | 36.88 | 52.50 | 63.18 | 39.30 | 52.83 | 58.80 | 60.65 | 76.11 | 84.06 |
| LMCL | 35.20 | 53.14 | 61.24 | 37.15 | 49.41 | 55.02 | 55.15 | 71.51 | 84.17 |
| Softmax | 34.68 | 53.10 | 62.61 | 33.56 | 44.22 | 53.26 | 54.25 | 72.36 | 82.65 |
| ADB | 52.22 | 56.64 | 65.58 | 35.14 | 45.54 | 55.36 | 77.51 | 77.92 | 81.97 |
| DCL | 47.63 | 50.15 | 54.11 | 31.1 | 38.22 | 45.55 | 76.31 | 81.92 | 83.79 |
| **Ours** | **61.43** | **64.54** | **68.25** | **48.82** | **58.51** | **63.32** | **78.05** | **83.74** | **84.99** |
| dataset | CLINC150 | | | Banking | | | StackOverflow | | |
| p=10% | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 | 0.25 | 0.5 | 0.75 |
| MSP | 53.77 | 71.50 | 77.39 | 41.97 | 48.21 | 68.69 | 45.89 | 61.02 | 78.29 |
| DOC | 13.21 | 22.57 | 54.57 | 10.39 | 20.33 | 24.84 | 43.69 | 60.43 | 61.60 |
| SEG | 68.29 | 77.52 | 80.34 | 56.75 | 58.69 | 71.61 | 59.24 | 78.64 | 83.85 |
| LMCL | 66.40 | 76.47 | 79.05 | 53.77 | 63.91 | 68.07 | 55.40 | 77.26 | 85.84 |
| Softmax | 64.59 | 77.01 | 79.72 | 52.70 | 60.42 | 69.31 | 57.24 | 77.48 | 84.43 |
| ADB | 67.49 | 74.82 | 77.76 | 50.04 | 66.01 | 70.75 | 67.41 | 81.08 | 84.62 |
| DCL | 67.99 | 72.55 | 70.76 | 54.02 | 61.27 | 65.98 | 75.99 | 82.09 | 84.52 |
| **Ours** | **71.93** | **78.06** | **82.19** | **59.76** | **67.73** | **74.09** | **78.91** | **83.82** | **85.93** |

Table 4: Macro f1-score excluding the OOD class for few-shot OOD intent detection with different proportion (0.25, 0.5 and 0.75) of in-distribution classes. $p$ indicates the ratio of selected few-shot in-distribution examples. For each setting, the best result is marked in bold.

### 5.2 Experimental Setup

To achieve a fair comparison, all the baselines and our method use the same pre-trained BERT model (bert-base-uncased (Wolf et al., 2019)) to encode input sentences.

To construct few-shot OOD intent detection tasks from the three datasets, we randomly sample 5% and 10% labeled examples per class as the training set from each of the three datasets. Then, we randomly select 25%, 50%. 75% of the classes in each dataset as in-distribution (ID) classes and set aside the respective remaining classes to the OOD class for the test stage. Concrete numbers of ID examples per class for each dataset can be found in Table 2. In particular, during training and validation, only the labeled few-shot examples of ID classes are seen by the model.

At training stage I, we use a two-layer MLP as $q_\theta$ and optimize the parameters of $q_\theta$ by Adam (Kingma and Ba, 2015) with a learning rate of $1e^{-4}$. The dropout rate for the corruption function is set to be 0.3 for all experiments. At training stage II, we instantiate our (k+1)-way OOD intent classifier $f_\phi$ by removing the pre-trained heads of BERT and appending a single layer MLP. For optimizing $f_\phi$, we adopt AdamW (Wolf et al., 2019) as optimizer and set the learning rate as $2e^{-5}$ following common practice (Devlin et al., 2019).

For the synthetic ID examples, we sample 15 reconstructed examples per real ID example. For the simulated OOD samples, we construct 100 OOD examples per batch during training. These values are selected with respect to the performance on validation sets. The reported results are the mean of 5 runs with different random seeds.

Following previous works (Yan et al., 2020; Zhang et al., 2021b; Zhan et al., 2021) in OOD intent detection, we use macro f1-score as the primary evaluation metric.

### 5.3 Correctness of the Synthetic In-distribution Examples

In Figure 4, we provide a qualitative evaluation of the generated synthetic in-distribution (ID) examples using t-SNE visualization (Van der Maaten and Hinton, 2008). We use the BERT embeddings of 5% labeled examples of 8 ID classes and all out-of-distribution examples from CLINC150 and plot them on the top of the figure. By generating 10 synthetic ID examples for each real ID example, we have the bottom figure where we can observe that these synthetic ID examples closely situate in

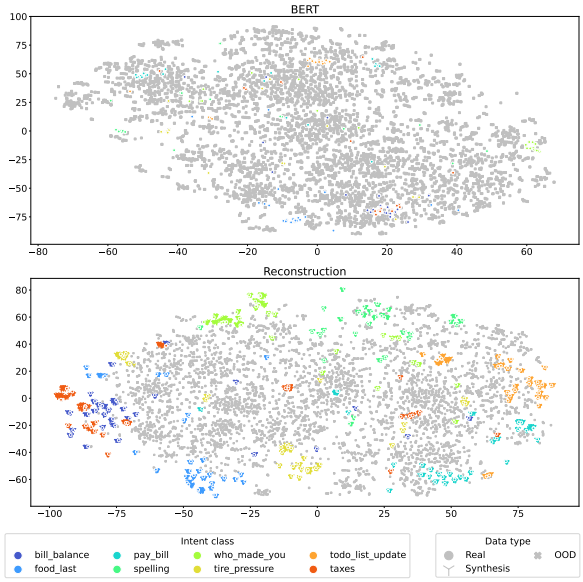


Figure 4: t-SNE visualization of BERT embeddings. Top: BERT embeddings without the synthetic in-distribution examples; Bottom: BERT embeddings with the synthetic in-distribution examples. Better view in color and enlarged.

the vicinity of each real ID example. Since BERT embeddings have been proved to be rich in contextualized semantics (Devlin et al., 2019), the distance between different embeddings can reflect the semantic gap between them. In this regard, at a high level, our generated ID examples can capture the expressiveness of ID classes.

### 5.4 Main Results

We present the results for the aforementioned three datasets in Table 3 and Table 4. As shown in the two tables, our proposed method consistently outperforms all baselines by a large margin in all settings.

Table 3 presents the results in overall macro f1-score on $(K+1)$ classes including the OOD class. The results in this table can be interpreted as the overall performance of the model. We first inspect the challenging case, where only $5\%$ labeled examples per class are sampled for training as shown in the top of Table 3. We can observe that our method leads to large improvements on all three datasets. In the most challenging case (only $25\%$ of classes in each dataset are selected as in-distribution classes), the improvement is more than $\mathbf{9}\%$ on CLINC150 and $\mathbf{8}\%$ on Banking than the second best results. Moreover, in the $50\%$ and $75\%$ cases, the improvements are also significant. For example, in the $50\%$ case of Banking, the gap between our method

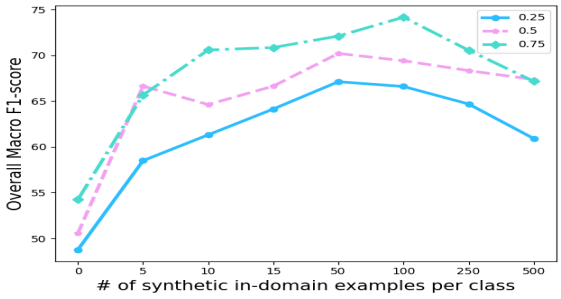


Figure 5: Effect of the number of synthetic in-distribution examples.

and the second best one is around $6.6\%$. These results verify the effectiveness and consistency of our model in extreme data-scarcity scenarios. As the ratio of labeled examples per class increased to $p = 10\%$, it can be seen that the baselines are improved by a large margin compared with the case of $p = 5\%$. However, our method can still achieve consistent improvement. This validates the robustness of our method under various data-scarcity scenarios.

In Table 4, we summarize the results in macro f1-score of in-distribution classes to demonstrate the effectiveness of synthetic ID examples in our method. It can be seen that in all settings, the performance gains are consistent with the results in Table 3, which indicates that the synthetic ID examples sampled from the DAE can help to improve the classification performance on ID classes.

| CLINC150, p=5% | | | |
|---|---|---|---|
| | Method | ID-F1 | Overall-F1 |
| 25% | SEG | 36.88 | 36.09 |
| | SEG + Ours | **63.65** | **64.25** |
| 50% | SEG | 52.50 | 51.90 |
| | SEG + Ours | **71.97** | **72.13** |
| 75% | SEG | 63.18 | 62.64 |
| | SEG + Ours | **70.67** | **70.72** |

Table 5: Results of SEG (Yan et al., 2020) and SEG with our synthetic ID examples (SEG + Ours). ID-F1 stands for in-distribution f1-score, and overall-F1 indicates the macro f1-score for all classes including the OOD class. Better results are marked in bold.

### 5.5 Effectiveness of the Synthetic In-distribution Examples

First, we study the impact of the number of synthetic in-distribution (ID) examples. We conduct experiments on the $5\%$ labeled ratio case. As shown in Figure 5, we vary the number of syn-

thetic ID examples per class from 0 to 500. In the range of [0,100], the classification performance increases gradually for all cases (0.25, 0.5 and 0.75). It shows the expressiveness of the synthetic ID examples. However, in the range of [100,500], we observe a slow performance drop in all cases. This is probably because the ID generator is learned from few-shot data and may generate inaccurate ID examples.

To further verify the effectiveness of our synthetic generator, we incorporate the synthetic ID examples to a strong baseline SEG (Yan et al., 2020) and present the results under the $p = 5\%$ setting of CLINC150 in Table 5. With our enlarged ID training set, the performance of SEG can also be improved significantly.

### 5.6 Robustness of the $(K+1)$-way Training Paradigm

In this subsection, we conduct experiments to evaluate the robustness of the $(K+1)$-way training paradigm with synthetic in-distribution (ID) examples.

As shown in Figure 6, we vary the corruption rate (from $0\%$ to $100\%$) of the learned latent denoising autoencoder (DAE) (trained by $30\%$ corruption rate). Notice that $100\%$ corruption rate indicates that no useful reconstruction information is passed to the DAE. We can observe that in the $0.5$ (orange line) and $0.75$ (green line) cases, the learned $(K+1)$-way classifier can maintain a surprisingly consistent performance compared with the $0.25$ (purple line). Especially, with $90\%$ corruption rate, the synthetic in-distribution (ID) examples are much less accurate than those with $30\%$ or $40\%$ corruption rate, but the performance does not drop to an unacceptable level. This verifies the noise neutralization effect of the $(K+1)$-way training manner discussed in Section 4.

## 6 Conclusion

In this paper, we have investigated few-shot OOD intent detection under a more challenging setting. We have conducted a pilot study to identify the key challenge for this problem, which is in improving the in-distribution (ID) expressiveness during training. To this end, we have proposed a promising approach to enrich the ID training set by sampling from a denoising autoencoder trained with only a few examples. The enlarged training set enables to train a well-performing $(K+1)$-way classifier. Our proposed approach has been validated by extensive experiments on real-world bechmarks.

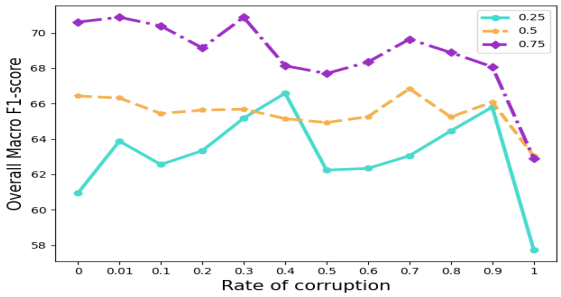


Figure 6: Effect of the rate of corruption on the learned denoising autoencoder. The experiment is conducted on CLINC150 under the $p = 5\%$ setting.

## Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This research was supported by the grants HK ITF UIM/377 and ITS/359/21FP.

## References

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *CoRR*, abs/2003.04807.

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.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. 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 55–65.

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.

Taewon Jeong and Heeyoung Kim. 2020. Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification. *Advances in Neural Information Processing Systems*, 33.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Seonghan Ryu, Seokhwan Kim, Junhwi Choi, Hwanjo Yu, and Gary Geunbae Lee. 2017. Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems. *Pattern Recogn. Lett.*, 88(C):26–32.

Seonghan Ryu, Sangjun Koo, Hwanjo Yu, and Gary Geunbae Lee. 2018. Out-of-domain detection based on generative adversarial network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 714–718, Brussels, Belgium. Association for Computational Linguistics.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Kuan-Chieh Wang, Paul Vicol, Eleni Triantafillou, and Richard Zemel. Few-shot out-of-distribution detection.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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, VS@NAACL-HLT 2015, June 5, 2015, Denver, Colorado, USA*, pages 62–69. The Association for Computational Linguistics.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y.S. 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, Online. Association for Computational Linguistics.

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. 2021a. Deep open intent classification with adaptive decision boundary. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14374–14382.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021b. Deep open intent classification with adaptive decision boundary. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14374–14382.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5064–5082. Association for Computational Linguistics.