Estimating Soft Labels for Out-of-Domain Intent Detection

Hao Lang * **Yinhe Zheng** * † **Jian Sun** **Fei Huang** **Luo Si** **Yongbin Li** †
Alibaba Group
{hao.lang, jian.sun, f.huang, luo.si,shuide.lyb}@alibaba-inc.com,
zhengyinhe1@163.com

Abstract

Out-of-Domain (OOD) intent detection is important for practical dialog systems. To alleviate the issue of lacking OOD training samples, some works propose synthesizing pseudo OOD samples and directly assigning one-hot OOD labels to these pseudo samples. However, these one-hot labels introduce noises to the training process because some "hard" pseudo OOD samples may coincide with In-Domain (IND) intents. In this paper, we propose an adaptive soft pseudo labeling (ASoul) method that can estimate soft labels for pseudo OOD samples when training OOD detectors. Semantic connections between pseudo OOD samples and IND intents are captured using an embedding graph. A co-training framework is further introduced to produce resulting soft labels following the *smoothness assumption*, i.e., close samples are likely to have similar labels. Extensive experiments on three benchmark datasets show that ASoul consistently improves the OOD detection performance and outperforms various competitive baselines.

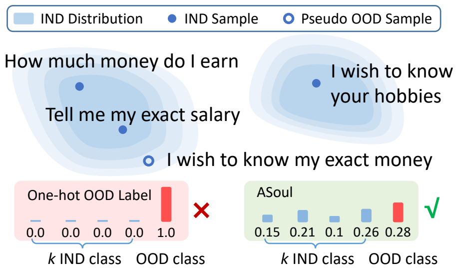

Figure 1: A pseudo OOD sample generated by distorting IND inputs (See more examples in Appendix A). Comparing to the one-hot OOD label, the soft label produced by ASoul is more suitable for this pseudo OOD sample since it carries some IND intents.

1 Introduction

Intent detection is essential for dialogue systems, and current methods usually achieve high performance under the *closed-world assumption* (Shu et al., 2017), i.e., data distributions are static, and only a fixed set of intents are considered. However, such an assumption may not be valid in practice, where we usually confront an *open-world* (Fei and Liu, 2016), i.e., unknown intents that are not trained may emerge. It is necessary to equip dialogue systems with Out-of-Domain (OOD) detection abilities so that they can accurately classify known In-Domain (IND) intents while rejecting unknown OOD intents (Yan et al., 2020a; Shen et al., 2021).

A major challenge for OOD detection is the lack of OOD samples (Xu et al., 2020a). In most applications, it is hard, if not impossible, to collect OOD samples from the test distribution before training (Du et al., 2021). To tackle this issue, various studies try to synthesize pseudo OOD samples in the training process. Existing methods include distorting IND samples (Choi et al., 2021; Shu et al., 2021; Ouyang et al., 2021), using generative models (Ryu et al., 2018; Zheng et al., 2020a), or even mixing-up IND features (Zhou et al., 2021a; Zhan et al., 2021). Promising results are reported by training a $(k+1)$-way classifier (k IND classes + 1 OOD class) using these pseudo OOD samples (Geng et al., 2020). This classifier can classify IND intents while detecting OOD intent since inputs that fall into the OOD class are regarded as OOD inputs.

Previous studies directly assign one-hot OOD labels to pseudo OOD samples when training the $(k+1)$-way classifier (Shu et al., 2021; Chen and Yu, 2021). However, this scheme brings noise to the training process because "hard" pseudo OOD samples, i.e., OOD samples that are close to IND distributions, may carry IND intents (Zhan et al., 2021) (See Figure 1). Indiscriminately assigning one-hot OOD labels ignores the semantic connections between pseudo OOD samples and IND intents. Moreover, this issue becomes more severe

* Equal contribution.
† Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 261–276
December 7-11, 2022 ©2022 Association for Computational Linguistics

as most recent studies are dedicated to producing hard pseudo OOD samples (Zheng et al., 2020a; Zhan et al., 2021) since these samples are reported to facilitate OOD detectors better (Lee et al., 2017). Collisions between pseudo OOD samples and IND intents will be more common.

We argue that ideal labels for pseudo OOD samples should be soft labels that allocate non-zero probabilities to all intents (Hinton et al., 2015; Müller et al., 2019). Specifically, we demonstrate in §3.2 that pseudo OOD samples generated by most existing approaches should be viewed as unlabeled data since they may carry both IND and OOD intents. Soft labels help capture the semantic connections between pseudo OOD samples and IND intents. Moreover, using soft labels also conforms to the *smoothness assumption*, i.e., samples close to each other are likely to receive similar labels. This assumption lays a foundation for modeling unlabeled data in various previous works (Luo et al., 2018; Van Engelen and Hoos, 2020).

In this study, we propose an adaptive soft pseudo labeling (ASoul) method that can estimate soft labels for given pseudo OOD samples and thus help to build better OOD detectors. Specifically, we first construct an embedding graph using supervised contrastive learning to capture semantic connections between pseudo OOD samples and IND intents. Following the smoothness assumption, a graph-smoothed label is produced for each pseudo OOD sample by aggregating nearby nodes on the graph. A co-training framework with two separate classification heads is introduced to refine these graph-smoothed labels. Concretely, the prediction of one head is interpolated with the graph-smoothed label to produce the soft label used to enhance its peer head. The final OOD detector is formulated as a $(k+1)$-way classifier with adaptive decision boundaries.

Extensive experiments on three benchmark datasets demonstrate that ASoul can be used with a wide range of OOD sample generation approaches and consistently improves the OOD detection performance. ASoul also helps achieve new State-of-the-art (SOTA) results on benchmark datasets. Our major contributions are summarized:

1. We propose ASoul, a method that can estimate soft labels for given pseudo OOD samples. ASoul conforms to the important smoothness assumption for modeling unlabeled data by assigning similar labels to close samples.

2. We construct an embedding graph to help capture the semantic connections between pseudo OOD samples and IND intents. A co-training framework is further introduced to produce the resulting soft labels with the help of two separate classification heads.

3. We conduct extensive experiments on three benchmark datasets. The results show that ASoul consistently improves the OOD detection performance, and it obtains new SOTA results.

2 Related Work

OOD Detection: OOD detection problems have been widely investigated in conventional machine learning studies (Geng et al., 2020). Recent neural-based methods try to improve the OOD detection performance by learning more robust representations on IND data (Zhou et al., 2021c, 2022; Yan et al., 2020b; Zeng et al., 2021b). These representations can be used to develop density-based or distance-based OOD detectors (Lee et al., 2018b; Podolskiy et al., 2021; Liu et al., 2020; Tan et al., 2019). Some methods also propose to distinguish OOD inputs using thresholds based methods (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Ren et al., 2019; Gangal et al., 2020; Ryu et al., 2017), or utilizing unlabeled IND data (Xu et al., 2021; Jin et al., 2022).

Pseudo OOD Sample Generation: Some works try to tackle OOD detection problems by generating pseudo OOD samples. Generally, four categories of approaches are proposed: **1.** Phrase Distortion (Chen and Yu, 2021): OOD samples are generated by replacing phrases in IND samples; **2.** Feature Mixup (Zhan et al., 2021): OOD features are directly produced by mixing up IND features (Zhang et al., 2018); **3.** Latent Generation (Marek et al., 2021): OOD samples are drawn from the low-density area of a latent space; **4.** Open-domain Sampling (Hendrycks et al., 2018): data from other corpora are directly used as pseudo OOD samples. With these pseudo OOD samples, the OOD detection task can be formalized into a $(k+1)$-way classification problem (k is the number of IND intents). Our method can be combined with all the above OOD generation approaches to improve the OOD detection performance.

Soft Labeling: Estimating soft labels for inputs has been applied in a wide range of studies such as knowledge distillation (Hinton et al., 2015; Gou et al., 2021; Zhang et al., 2020), confidence cal-

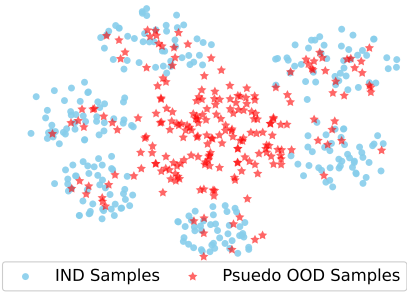

Figure 2: t-SNE visualization of pseudo OOD samples generated by feature mixup (Zhan et al., 2021) on the Banking dataset under the 25% setting. It can be seen that some pseudo OOD samples coincide with IND samples. See more analyses in Appendix A.

ibration (Müller et al., 2019; Wang et al., 2021), or domain shift (Ng et al., 2020). However, few studies try to utilize this approach in OOD detection methods. Existing approaches only attempt to assign dynamic weights (Ouyang et al., 2021) or soft labels to IND samples (Cheng et al., 2022). Our method ASoul is the first attempt to estimate soft labels for pseudo OOD samples.

Semi-Supervised Learning: Our work is also related to semi-supervised learning (SSL) since they all attempt to utilize unlabeled data and share the same underlying *smoothness assumption* (Wang and Zhou, 2017; Lee et al., 2013; Li et al., 2021). Moreover, the co-training framework in ASoul also helps to enforce the *low-density assumption* (a variant of the smoothness assumption) (Van Engelen and Hoos, 2020; Chen et al., 2022) by exploring low-density regions between classes.

3 Background

3.1 Problem Definition

OOD intent detectors aim to classify IND intents while detecting OOD inputs. Concretely, given k known IND intent classes $\mathcal{I} = \{I_i\}_{i=1}^k$, the training set $\mathcal{D}_I = \{(x_i, y_i)\}$ only contains IND samples, i.e., x_i is an input, and $y_i \in \mathcal{I}$ is the label of x_i. The test set $\bar{\mathcal{D}} = \{(\bar{x}_i, \bar{y}_i)\}$ consists both IND and OOD samples, i.e., $\bar{y}_i \in \mathcal{I} \cup \{I_{k+1}\}$, in which I_{k+1} is a special OOD intent class. For a testing input $\bar{x}_i$, an OOD detector should classify the intent of $\bar{x}_i$ if $\bar{x}_i$ belongs to an IND intent or reject $\bar{x}_i$ if $\bar{x}_i$ belongs to the OOD intent.

3.2 Analyzing Pseudo OOD Samples

Recent works have demonstrated that "hard" OOD samples, i.e., OOD samples akin to IND distributions, are more efficient in improving the OOD detection performance (Lee et al., 2018a; Zheng et al., 2020a). Promising performances are obtained using these hard samples on various benchmarks (Zhan et al., 2021; Shu et al., 2021).

However, we notice that hard pseudo OOD samples used in previous approaches may coincide with IND samples and carry IND intents. Besides Figure 1, we further demonstrate this issue by visualizing pseudo OOD samples produced by Zhan et al. (2021). Specifically, pseudo OOD samples are synthesized using convex combinations of IND features. Figure 2 shows the results on the Banking dataset (Casanueva et al., 2020) when 25% intents are randomly selected as IND intents. It can be seen that some pseudo OOD samples fall into the cluster of IND intents, and thus it is improper to assign one-hot OOD labels to these samples.

The above issue is also observed in other pseudo OOD sample generation approaches. Specifically, we implement the phrase distortion approach proposed by Shu et al. (2021) and employ crowd-sourced workers to annotate 1,000 generated pseudo OOD samples. Results show that up to 39% annotated samples carry IND intents (see Appendix A for more examples).

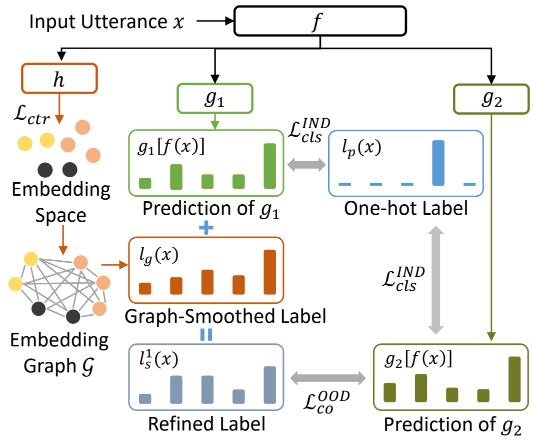

Figure 3: An overview of ASoul. Specifically, an embedding space is obtained using an encoder f and a projection head h by optimizing a supervised constructive loss $\mathcal{L}_{ctr}$ on labeled IND data. A graph-smoothed label $l_g(x)$ conforming to the smoothness assumption is constructed. $l_g(x)$ is further used in a co-training framework, in which two classification heads g_1 and g_2 are maintained. The prediction of one head is interpolated with $l_g(x)$ to enhance another head.

4 Method

4.1 Overview

In this study, we build the OOD intent detector following three steps: 1. Construct a set of pseudo OOD samples $\mathcal{D}_P$; 2. Estimate a soft label for each sample $x \in \mathcal{D}_P$; 3. Obtain a $(k+1)$-way classifier and learn a decision boundary for each class to build an OOD detector. A testing input x is identified as OOD if x belongs to the OOD intent I_{k+1} or x is out of all decision boundaries.

Before applying ASoul, we assume a set of pseudo OOD samples D_P are already generated using existing approaches. Figure 3 shows an overview of ASoul. Specifically, a shared utterance encoder f encodes each input $x \in \mathcal{D}_I \cup \mathcal{D}_P$ into a representation, and an embedding projection head h constructs an embedding graph on these representations. A co-training framework is also implemented using two $(k+1)$-way classification heads g_1 and g_2, and the prediction of one head is used to enhance soft labels of the peer head.

Note that ASoul is independent of specific methods to produce pseudo OOD samples in $\mathcal{D}_P$. In this study, we test various approaches to obtain $\mathcal{D}_P$.

4.2 Embedding Graph

Embedding Space: An embedding space is maintained in ASoul to capture semantic of input samples. Specifically, for an input x_i, an encoder f is used to convert x_i into a representation vector, then a projection head h is used to map $f(x_i)$ into an L2 normalized embedding $z_i = h[f(x_i)]$ to construct the embedding space. To capture better semantic representations, a supervised contrastive loss (Khosla et al., 2020; Gunel et al., 2020) $\mathcal{L}_{ctr}$ is optimized on labeled IND samples in $\mathcal{D}_I$:

$$\mathcal{L}_{ctr} = \sum_{x_i \in D_I} \frac{-1}{|S(i)|} \sum_{x_j \in S(i)} \log \frac{e^{\Phi(x_i)\cdot\Phi(x_j)/t}}{\sum\limits_{x_k \in A(i)} e^{\Phi(x_i)\cdot\Phi(x_k)/t}}, \tag{1}$$

in which $S(i)$ represents samples that share the same label with x_i in the current batch, $A(i)$ represents all samples in the current batch except x_i, Φ maps an input x to its corresponding embedding (i.e., $\Phi(x) = h[f(x)]$), and $t > 0$ is a scalar that controls the separation of classes. $\mathcal{L}_{ctr}$ captures the similarities between examples in the same class and contrast them with the examples from different classes (Gunel et al., 2020).

Graph-Smoothed Label: After obtaining the embedding space, we construct a fully connected unidirectional embedding graph $\mathcal{G}$ using samples in $\mathcal{D}_{IP} = \mathcal{D}_I \cup \mathcal{D}_P$. Specifically, we first map each sample $x \in \mathcal{D}_{IP}$ into an embedding z, i.e., $z = \Phi(x)$, and then use all these embeddings as nodes for $\mathcal{G}$. Every two nodes z_i and z_j in $\mathcal{G}$ are linked with an edge. Moreover, we also assign a *prior label* $l_p(x) \in \mathbb{R}^{k+1}$ to each sample $x \in \mathcal{D}_{IP}$ to represent its annotation, i.e., for an IND sample $x \in \mathcal{D}_I$, $l_p(x)$ is defined as the one-hot label corresponding to y, and for a pseudo OOD sample $x \in \mathcal{D}_P$, $l_p(x)$ is defined as the one-hot OOD label corresponding to I_{k+1}.

For each OOD sample $x \in \mathcal{D}_P$, a graph-smoothed label $l_g(x)$ is obtained by aggregating adjacent nodes on $\mathcal{G}$. Specifically, to conform to the smoothness assumption, we try to minimize the following distance when determining $l_g(x)$:

$$\alpha \cdot d[l_g(x), l_p(x)] + (1-\alpha) \sum_{x_j \in \mathcal{D}_{IP}} a_j \cdot d[l_g(x), l_p(x_j)]$$
$$a_j = \frac{\exp(z \cdot z_j/\tau)}{\sum_{k=1}^{|\mathcal{D}_{IP}|} \exp(z \cdot z_k/\tau)}, \tag{2}$$

where $0 \leq \alpha \leq 1$ is a scalar, d is a distance function, $\tau > 0$ is a scalar temperature. The second term in Eq. 2 enforces the smoothness assumption by encouraging $l_g(x)$ to have similar labels with its nearby samples, whereas the first term tries to maintain $l_g(x)$ to meet its original annotation $l_p(x)$. For simplicity, we implement d as the Euclidean distance here, and thus minimizing Eq. 2 yields:

$$l_g(x) = \alpha \cdot l_p(x) + (1-\alpha) \sum_{x_j \in \mathcal{D}_{IP}} a_j \cdot l_p(x_j) \tag{3}$$

Note that the result we derived in Eq. 3 follows most previous graph-smoothing approaches in semi-supervised learning (Van Engelen and Hoos, 2020). To the best of our knowledge, we are the first to apply this scheme to OOD detection tasks.

4.3 Co-Training Framework

To further enforce the smoothness assumption, a co-training framework is introduced in ASoul to learn better soft labels using $l_g(x)$. Specifically, we implement two classification heads g_1 and g_2 on top of the shared encoder f. Each classification head g_i maps the output of f to a $(k+1)$ dimensional distribution, i.e, $g_i[f(x)] \in \mathbb{R}^{k+1}$ $(i = 1, 2)$, and a classification loss is optimized on IND samples:

$$\mathcal{L}_{cls}^{IND} = \sum_{x \in \mathcal{D}_I} \frac{1}{2} \sum_{i=1}^{2} CE(l_p(x), g_i[f(x)]), \tag{4}$$

in which CE measures the cross-entropy between two distributions.

Besides optimizing $\mathcal{L}_{cls}^{IND}$, a co-training process is implemented to refine $l_g(x)$ for each $x \in \mathcal{D}_P$. Specifically, a soft label $l_s^1(x)$ (or $l_s^2(x)$) is produced by interpolating $l_g(x)$ with the prediction of one classification head g_1 (or g_2), and the resulting soft label is used to optimize another head g_2 (or g_1). Concretely, the following co-training loss is optimized:

$$\mathcal{L}_{co}^{OOD} = \sum_{x \in \mathcal{D}_P} \frac{1}{2} \sum_{j=1}^{2} \sum_{i=1}^{2} \mathbf{1}_{i \neq j} CE(l_s^i(x), g_j[f(x)]), \quad (5)$$
$$l_s^i(x) = \beta \cdot l_g(x) + (1-\beta) \cdot g_i[f(x)], (i = 1, 2),$$

where $0 \leq \beta \leq 1$ is a weight scalar. Different dropout masks are used in g_1 and g_2 to prompt the diversity required by co-training. Note that as indicated by Lee et al. (2013) and Chen et al. (2022), the co-training loss $\mathcal{L}_{co}^{OOD}$ favors low density separation between classes, and thus it helps to enforce the low-density assumption when training g_i.

The overall training loss for our method is:

$$\mathcal{L} = \mathcal{L}_{ctr} + \mathcal{L}_{cls}^{IND} + \mathcal{L}_{co}^{OOD} \quad (6)$$

4.4 OOD Detection

In the inference phase, we directly use the averaged prediction of g_1 and g_2 to implement the OOD detector $g(y|x) \in \mathbb{R}^{k+1}$.

$$g(y|x) = (g_1[f(x)] + g_2[f(x)])/2 \quad (7)$$

Moreover, an adaptive decision boundary (ADB) is learnt on top of $g(y|x)$ to further reduce the open space risk (Zhou et al., 2022; Shu et al., 2021). Specifically, we follow the approach of Zhang et al. (2021) to obtain a central vector c_i and a decision boundary scalar b_i for each intent class $I_i \in \mathcal{I} \cup \{I_{k+1}\}$. In the testing phase, the label y for each input x is obtained as:

$$y = \begin{cases} I_{k+1}, & \text{if } ||f(x) - c_i|| > b_i, \forall i \in \{1, \dots, k+1\}, \\ \underset{I_j \in \mathcal{I} \cup \{I_{k+1}\}}{\arg\max} \; g(I_j|x), & \text{otherwise}, \end{cases}$$

In this way, we can classify IND intents while rejecting OOD intent.

5 Experiments

5.1 Datasets

Following most previous works (Zhang et al., 2021; Shu et al., 2021), we use three benchmark datasets:

Dataset	Train	Valid	Test	#Intent
CLINC150	15,000	3,000	5,700	150
StackOverflow	12,000	2,000	6,000	20
Banking	9,003	1,000	3,080	77

Table 1: Dataset statistics.

CLINC150 (Larson et al., 2019) contains 150 IND intents and one OOD intent. We follow Zhan et al. (2021) to group all samples from the OOD intent into the test set; **StackOverflow** (Xu et al., 2015) contains 20 classes with 1,000 samples in each class; **Banking** (Casanueva et al., 2020) contains 77 intents in the banking domain. Standard splits of above datasets is followed (See Table 1).

5.2 Implementation Details

Our encoder f is implemented using BERT (Devlin et al., 2018) with a mean-pooling layer. The projection head h, classification heads $g1$ and g_2 are implemented as two-layer MLPs with the LeakyReLU activation (Xu et al., 2020b). The optimizer AdamW and Adam (Kingma and Ba, 2014) is used to finetune BERT and all the heads with a learning rate of 1e-5 and 1e-4, respectively. We use $\tau = 0.1$, $\alpha = 0.11$ and $\beta = 0.9$ in all experiments. All results reported in our paper are averages of 10 runs with different random seeds. See Appendix B for more implementation details. Note that ASoul only introduces little computational overhead compared to the vanilla BERT model (See Appendix D.), and we detail how to choose important hyper-parameters for ASoul in Appendix E.

5.3 Experiment Setups and Baselines

Following (Zhang et al., 2021; Zhan et al., 2021; Shu et al., 2021), we randomly sample 25%, 50%, and 75% intents as the IND intents and regard all remaining intents as one OOD intent I_{k+1}. Note that in the training and validation process, we only use samples from the IND intents. Hyper-parameters are searched based on IND intent classification performances on validation sets.

To validate our claim that ASoul is independent of specific methods to produce $\mathcal{D}_P$. We tested the performance of ASoul with four pseudo OOD sample generation approaches: 1. *Phrase Distortion* (**PD**): follows Shu et al. (2021) to generates OOD samples by distorting IND samples; 2. *Feature Mixup* (**FM**): follows Zhan et al. (2021) to produce OOD features using convex combinations of

	Methods	CLINC150 Acc-All	CLINC150 F1-All	CLINC150 F1-OOD	CLINC150 F1-IND	StackOverflow Acc-All	StackOverflow F1-All	StackOverflow F1-OOD	StackOverflow F1-IND	Banking Acc-All	Banking F1-All	Banking F1-OOD	Banking F1-IND
25%	LG+Onehot	30.54	39.16	24.25	39.55	42.45	43.20	44.43	42.95	26.72	48.04	9.03	50.09
	OS+Onehot	45.56	47.04	48.81	47.00	26.93	38.28	8.89	44.16	29.86	44.09	14.92	45.62
	PD+Onehot	89.79	80.04	93.42	79.69	91.53	85.05	94.41	83.18	81.69	73.44	87.11	72.72
	FM+Onehot	89.78	82.04	93.34	81.74	89.76	76.95	93.60	73.62	79.42	72.70	84.87	72.06
	LG+ASoul	32.38	40.70	27.45	41.05	58.00	53.97	64.81	51.80	29.06	48.81	13.83	50.66
	OS+ASoul	50.67	49.70	55.80	49.54	27.23	41.29	9.66	47.61	33.38	47.34	22.14	48.67
	PD+ASoul	90.61	81.68	93.95	81.36	**93.20**	**87.01**	**95.58**	**85.30**	85.32	**78.64**	89.80	**78.06**
	FM+ASoul	**92.71**	**84.11**	**95.42**	**83.81**	92.04	86.03	94.76	84.29	**87.41**	78.39	**91.52**	77.70
50%	LG+Onehot	42.51	61.71	12.89	62.36	60.00	67.42	52.92	68.87	47.89	66.84	4.37	68.49
	OS+Onehot	55.44	66.88	43.80	67.19	47.92	60.26	7.93	65.49	49.98	66.82	11.82	68.26
	PD+Onehot	88.61	86.57	90.62	86.52	88.52	87.35	89.57	87.13	80.90	81.78	81.32	81.79
	FM+Onehot	87.91	87.06	89.71	87.03	83.47	80.78	85.48	80.31	80.32	81.48	80.57	81.50
	LG+ASoul	46.53	63.09	24.26	63.60	63.25	71.54	58.03	72.90	50.45	68.10	12.63	69.56
	OS+ASoul	58.19	68.46	49.04	68.72	49.13	62.52	12.11	67.56	50.61	68.52	12.33	70.00
	PD+ASoul	89.50	87.54	91.40	87.49	**89.45**	**88.65**	**90.46**	**88.47**	81.86	83.84	81.51	83.90
	FM+ASoul	**89.96**	**88.20**	**91.72**	**88.15**	88.92	88.24	89.69	88.09	**81.98**	**83.96**	**81.65**	**84.03**
75%	LG+Onehot	57.82	74.64	7.47	75.24	69.58	76.84	16.52	80.86	72.31	82.34	13.32	83.53
	OS+Onehot	69.54	80.00	47.04	80.29	68.81	75.33	5.39	80.00	71.48	81.66	7.97	82.92
	PD+Onehot	87.70	89.30	85.86	89.33	83.75	86.88	75.21	87.66	82.79	86.94	71.95	87.20
	FM+Onehot	88.59	89.66	87.18	89.68	83.44	86.67	73.62	87.53	82.06	87.31	65.60	87.68
	LG+ASoul	59.70	75.62	13.06	76.19	71.07	77.76	20.09	81.61	73.64	83.51	19.58	84.61
	OS+ASoul	71.60	81.25	52.20	81.51	69.51	76.67	8.24	81.23	72.23	82.44	10.97	83.68
	PD+ASoul	88.59	90.57	86.45	90.60	84.53	87.44	73.75	88.35	83.30	87.76	69.03	88.09
	FM+ASoul	**89.88**	**91.38**	**88.21**	**91.41**	**85.00**	**87.90**	**75.76**	**88.71**	**84.47**	**88.39**	**72.64**	**88.66**

Table 2: Performances of ASoul when combined with different OOD sample generation approaches. Best results among each setting are bolded. The best performing ASoul-based method significantly outperforms other baselines with p-value < 0.05 (t-test) in each setting.

IND features; 3. *Latent Generation* (**LG**): follows Zheng et al. (2020a) to decode pseudo OOD samples from a latent space; 4. *Open-domain Sampling* (**OS**): follows Zhan et al. (2021) to use sentences from other corpora as OOD samples. Each approach mentioned above associates with one of the four categories listed in §2.

Moreover, we also applied the above pseudo OOD sample generation approaches with the previous SOTA method that uses one-hot labeled pseudo OOD samples (Shu et al., 2021). Specifically, a $(k+1)$-way classifier is trained by optimizing the cross-entropy loss on $\mathcal{D}_I \cup \mathcal{D}_P$ using one-hot labels, and the ADB approach presented in §4.4 is used to construct the OOD detector.

We also compared our method to other competitive OOD detection baselines: **MSP:** (Hendrycks and Gimpel, 2017) utilizes the maximum Softmax probability of a k-way classifier to detect OOD inputs; **DOC:** (Shu et al., 2017) employs k 1-vs-rest Sigmoid classifiers and use the maximum predictions to detect OOD intents; **OpenMax:** (Bendale and Boult, 2016) fits a Weibull distribution to the logits and re-calibrates the confidences with an Open-Max Layer; **LMCL:** (Lin and Xu, 2019) introduces a large margin cosine loss to maximize the decision margin and uses LOF as the OOD detector; **ADB:** (Zhang et al., 2021) learns an adaptive decision boundaries for OOD detection; **Outlier:** (Zhan et al., 2021) mixes convex interpolated outliers and open-domain outliers to train a $(k+1)$-way classifier; **SCL:** (Zeng et al., 2021a) uses a supervised contrastive learning loss to separate IND and OOD features; **GOT:** (Ouyang et al., 2021) shapes an energy gap between IND and OOD samples. **ODIST:** (Shu et al., 2021) generates pseudo OOD samples with using a pre-trained language model.

For fair comparisons, all baselines are implemented with codes released by their authors, and use BERT as the backbone. For threshold-based baselines, 100 OOD samples are used in the validation to determine the thresholds used for testing. See Appendix C for more details about baselines.

5.4 Metrics

Following Zhang et al. (2021); Zhan et al. (2021); Shu et al. (2021), we use overall accuracy (**Acc-All**) and macro F1-score (**F1-All**) calculated over

	Methods	CLINC150 Acc-All	F1-All	F1-OOD	F1-IND	StackOverflow Acc-All	F1-All	F1-OOD	F1-IND	Banking Acc-All	F1-All	F1-OOD	F1-IND
25%	MSP	47.02	47.62	50.88	47.53	28.67	37.85	13.03	42.82	43.67	50.09	41.43	50.55
	DOC	74.97	66.37	81.98	65.96	42.74	47.73	41.25	49.02	56.99	58.03	61.42	57.85
	OpenMax	68.50	61.99	75.76	61.62	40.28	45.98	36.41	47.89	49.94	54.14	51.32	54.28
	SCL	75.01	65.45	81.92	65.01	62.08	61.01	67.99	59.61	70.82	64.82	77.28	64.17
	GOT	72.63	64.01	79.45	63.60	65.02	62.26	68.58	61.00	63.05	63.49	68.61	63.22
	LMCL	81.43	71.16	87.33	70.73	47.84	52.05	49.29	52.60	64.21	61.36	70.44	60.88
	ADB	87.59	77.19	91.84	76.80	86.72	80.83	90.88	78.82	78.85	71.62	84.56	70.94
	Outlier	88.44	80.73	92.35	80.43	68.74	65.64	74.86	63.80	74.11	69.93	80.12	69.39
	ODIST	89.79	80.04	93.42	79.69	91.53	85.05	94.41	83.18	81.69	73.44	87.11	72.72
	FM+ASoul	**92.71**	**84.11**	**95.42**	**83.81**	**92.04**	**86.03**	**94.76**	**84.29**	**87.41**	**78.39**	**91.52**	**77.70**
50%	MSP	62.96	70.41	57.62	70.58	52.42	63.01	23.99	66.91	59.73	71.18	41.19	71.97
	DOC	77.16	78.26	79.00	78.25	52.53	62.84	25.44	66.58	64.81	73.12	55.14	73.59
	OpenMax	80.11	80.56	81.89	80.54	60.35	68.18	45.00	70.49	65.31	74.24	54.33	74.76
	SCL	71.14	75.03	70.81	75.09	76.16	78.95	74.42	79.40	74.81	78.04	72.45	78.19
	GOT	67.06	73.15	63.48	73.28	65.56	72.19	55.53	73.86	69.97	76.37	63.03	76.72
	LMCL	83.35	82.16	85.85	82.11	58.98	68.01	43.01	70.51	72.73	77.53	69.53	77.74
	ADB	86.54	85.05	88.65	85.00	86.40	85.83	87.34	85.68	78.86	80.90	78.44	80.96
	Outlier	88.33	86.67	90.30	86.54	75.08	78.55	71.88	79.22	72.69	79.21	67.26	79.52
	ODIST	88.61	86.57	90.62	86.52	88.52	87.35	89.57	87.13	80.90	81.78	81.32	81.79
	FM+ASoul	**89.96**	**88.20**	**91.72**	**88.15**	**88.92**	**88.24**	**89.69**	**88.09**	**81.98**	**83.96**	**81.65**	**84.03**
75%	MSP	74.07	82.38	59.08	82.59	72.17	77.95	33.96	80.88	75.89	83.60	39.23	84.36
	DOC	78.73	83.59	72.87	83.69	68.91	75.06	16.76	78.95	76.77	83.34	50.60	83.91
	OpenMax	76.80	73.16	76.35	73.13	74.42	79.78	44.87	82.11	77.45	84.07	50.85	84.64
	SCL	76.50	82.65	66.90	82.79	79.91	84.41	63.79	85.78	78.45	84.09	56.19	84.57
	GOT	72.65	81.49	54.11	81.73	77.76	81.85	52.80	83.79	77.11	83.36	48.30	83.97
	LMCL	83.71	86.23	81.15	86.27	72.33	78.28	37.59	81.00	78.52	84.31	58.54	84.75
	ADB	86.32	88.53	83.92	88.58	82.78	85.99	73.86	86.80	81.08	85.96	66.47	86.29
	Outlier	88.08	89.43	86.28	89.46	81.71	85.85	65.44	87.22	81.07	86.98	60.71	87.47
	ODIST	87.70	89.30	85.86	89.33	83.75	86.88	75.21	87.66	82.79	86.94	71.95	87.20
	FM+ASoul	**89.88**	**91.38**	**88.21**	**91.41**	**85.00**	**87.90**	**75.76**	**88.71**	**84.47**	**88.39**	**72.64**	**88.66**

Table 3: Performance of ASoul and baselines. Best results among each setting are bolded. All improvements of our method over baselines are significant with p-value < 0.05 (t-test).

all intents (IND and OOD intents) to evaluate the OOD detection performance. We also calculate macro F1-scores over IND intents (**F1-IND**) and OOD intent (**F1-OOD**) to evaluate fine-grained performances.

5.5 Results

Table 2 shows the OOD detection performance associated with different pseudo OOD sample generation approaches. Specifically, results marked with "ASoul" measures the performance of our method, while results marked with "Onehot" correspond to the performance of the previous SOTA method (Shu et al., 2021) that uses one-hot labeled samples. We can observe that: **1.** ASoul consistently outperforms its one-hot labeled counterpart with large margins. This validates our claim that ASoul can be used to improve the OOD detection performance with different pseudo OOD sample generation approaches; **2.** "hard" pseudo OOD samples yield by **FM** lead to sub-optimal performance when assigned with one-hot labels (i.e., FM+Onehot generally under-performs PD+Onehot), while it achieves the best performance when combined with ASoul. This demonstrates that assigning one-hot labels to hard pseudo OOD samples introduces noise to the training process and ASoul helps to alleviate these noises. **3.** Although OOD samples yielded by the open-domain sampling approach are usually disjoint from the training task, they still benefit from ASoul. We suppose this is because the soft labels produced by ASoul prevent the OOD detector from becoming over-confident, which is important to improve the OOD detection performance.

Table 3 shows the performance of all baselines and our best method FM+ASoul. It can be seen that FM+ASoul significantly outperforms all baselines and achieve SOTA results on all three datasets. This validates the effectiveness of ASoul in improving the OOD detection performance. We can also observe large improvements of ASoul when labeled IND datasets are small (i.e., in 25% and 50% settings). This demonstrates the potential of ASoul to be applied in practical scenarios, particularly in the early phases of the development that we usually need to handle a large number of OOD inputs with limited IND intents (Zhan et al., 2021).

5.6 Ablation Study

Ablation studies were performed to verify the effect of each component in ASoul: We tested following variants: 1. **ASoul-CT** removes the co-training framework, i.e., only one classification head g_1 is implemented without the co-training process. In this variant, the loss shown in Eq.6 is optimized by moving g_2 and setting $\beta = 1$ in Eq.5. 2. **ASoul-GS** removes the graph-smoothed labels, i.e., the embedding graph is not constructed. In this variant, losses shown in Eq.4 and 5 are optimized and $l_g(x)$ in Eq.5 is replaced with the one-hot prior label $l_p(x)$. 3. **USoul** employs uniformly distributed soft labels for samples in $\mathcal{D}_P$. In this variant, the soft label $l_s^i(x)$ in Eq.5 is obtained by uniformly reallocating a small portion of OOD probability to OOD intents. 4. **KnowD** implements a knowledge distillation process to obtain soft labels, i.e., a k-way IND intent classifier is first trained on $\mathcal{D}_I$ and its predictions are interpolated with the one-hot OOD label to obtain the soft label $l_s^i(x)$ in Eq.5.

All above variants are tested with two approaches to produce $\mathcal{D}_P$: PD and FM. Results in Table 4 indicate that our method outperforms all ablation models. We can further observe that: **1.** soft-labels obtained using other approaches degenerate the model performance by a large margin. This shows the effectiveness of the soft labels produced by ASoul. **2.** graph-smoothed labels bring the largest improvement compared to other components. This further proves the importance of modeling semantic connections between OOD samples and IND intents.

5.7 Feature Visualization

To further demonstrate the effectiveness of ASoul, we visualized the features learnt in the penultimate layer of OOD detectors that are trained using one-hot labels or soft labels. We use the best performing pseudo OOD samples generation approach (i.e., FM) in this analysis. Results shown in Figure 4 demonstrate that soft labels produced by ASoul help the OOD detector learn better representations compared to one-hot labels. The learnt feature space is smoother and representations for IND and OOD samples are more separable. This validates our claim that ASoul helps to conform to the smoothness assumption and improves the OOD detection performance.

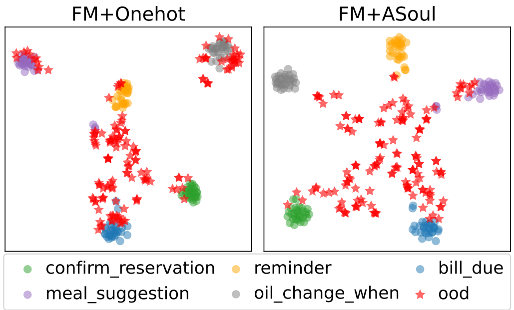

Figure 4: t-SNE visualization of learnt features on the test set of CLINC150 under the 25% setting.

6 Conclusion

In this paper, we first analyze the limitation of existing OOD detection approaches that use one-hot labeled pseudo OOD samples. Then we propose a method ASoul that can estimate soft labels for given pseudo OOD samples and use these soft labels to train better OOD detectors. An embedding space is constructed to produce graph-smoothed labels to capture the semantic connections between OOD samples and IND intents. A co-training framework further refines these graph-smoothed labels. Experiments demonstrate that our method can be combined with different pseudo OOD sample generation approaches, and it helps achieve SOTA results on three benchmark datasets. In the future, we plan to apply our method in other tasks, such as Text-to-SQL parsers (Hui et al., 2021; Wang et al., 2022; Qin et al., 2022) or lifelong learning (Dai et al., 2022).

Limitations

We identify the major limitation of this work is its input modality. Specifically, our method is limited to textual inputs and ignores inputs in other modalities such as vision, audio, or robotic features. These modalities provide valuable information that can be used to better OOD detectors. Fortunately, with the help of multi-modal pre-training models (Radford et al., 2021; Zheng et al., 2022), we can obtain robust features well aligned across different modalities. In future works, we will try to model multi-modal contexts for OOD detection and explore better pseudo OOD sample generation approaches.

Methods	25%				50%				75%			
	Acc-All	F1-ALL	F1-OOD	F1-IND	Acc-All	F1-ALL	F1-OOD	F1-IND	Acc-All	F1-ALL	F1-OOD	F1-IND
PD+ASoul	**90.61**	**81.68**	**93.95**	**81.36**	**89.50**	**87.54**	**91.40**	**87.49**	**88.59**	**90.57**	**86.45**	**90.60**
PD+ASoul-CT	90.42	81.36	93.82	81.04	88.36	87.05	90.20	87.01	88.28	90.14	86.16	90.18
PD+ASoul-GS	89.96	80.66	93.55	80.32	87.96	86.93	89.81	86.89	88.04	90.03	85.92	90.06
PD+USoul	89.33	80.20	93.05	79.87	87.72	86.59	89.60	86.55	87.21	89.29	85.02	89.33
PD+KnowD	89.49	79.46	93.33	79.10	87.09	85.51	89.05	85.47	85.79	88.11	83.20	88.16
FM+ASoul	**92.71**	**84.11**	**95.42**	**83.81**	**89.96**	**88.20**	**91.72**	**88.15**	**89.88**	**91.38**	**88.21**	**91.41**
FM+ASoul-CT	92.47	83.43	95.28	83.11	88.58	87.50	90.28	87.46	89.02	90.76	87.09	90.80
FM+ASoul-GS	91.73	82.92	94.77	82.61	88.98	87.31	90.73	87.27	88.82	90.50	86.96	90.53
FM+USoul	90.35	82.13	93.80	81.82	87.51	87.02	89.17	86.99	88.21	89.70	86.46	89.73
FM+KnowD	90.11	81.84	93.58	81.53	88.18	86.23	90.17	86.18	86.54	88.64	84.31	88.68

Table 4: Ablation study results on the CLINC150 dataset.

Another limitation of this work is the pre-training model used in experiments: a model pre-trained on dialogue corpora is expected to yield better performance (He et al., 2022c,a,b; Zhou et al., 2021b; Wang et al., 2020; Zheng et al., 2020b). Moreover, it is reported that better OOD detection performance can be obtained if we can extract more robust features for IND tasks (Vaze et al., 2021). Our method can be readily applied to other feature extractors that are better performed on dialogues.

Ethics Statement

This work does not present any direct ethical issues. In the proposed work, we seek to develop a general method for OOD intent detection, and we believe this study leads to intellectual merits that benefit from a reliable application of NLU models. All experiments are conducted on open datasets.

References

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.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Derek Chen and Zhou Yu. 2021. Gold: Improving out-of-scope detection in dialogues using data augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 429–442.

Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, and Chongjun Wang. 2022. Semi-supervised learning with multi-head co-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Cong Wang, and Qing Gu. 2022. Learning to classify open intent via soft labeling and manifold mixup. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin. 2021. Outflip: Generating examples for unknown intent detection with natural language attack. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 504–512.

Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, Luo Si, and Yongbin Li. 2022. Lifelong learning for question answering with hierarchical prompts. *arXiv preprint arXiv:2208.14602*.

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

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. 2021. Vos: Learning what you don't know by virtual outlier synthesis. In *International Conference on Learning Representations*.

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.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 2020. Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7764–7771.

Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. 2020. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Wanwei He, Yinpei Dai, Binyuan Hui, Min Yang, Zheng Cao, Jianbo Dong, Fei Huang, Luo Si, and Yongbin Li. 2022a. Space-2: Tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 553–569.

Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022b. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 187–200.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022c. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2018. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network.

Binyuan Hui, Ruiying Geng, Qiyu Ren, Binhua Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, Pengfei Zhu, and Xiaodan Zhu. 2021. Dynamic hybrid relation exploration network for cross-domain context-dependent semantic parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13116–13124.

Di Jin, Shuyang Gao, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tür. 2022. Towards textual out-of-domain detection without in-domain labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1386–1395.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 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.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2017. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018a. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018b. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Junnan Li, Caiming Xiong, and Steven CH Hoi. 2021. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484.

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.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. 2018. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8896–8905.

Petr Marek, Vishal Ishwar Naik, Anuj Goyal, and Vincent Auvray. 2021. Oodgan: Generative adversarial network for out-of-domain data generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 238–245.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283.

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

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13675–13682.

Bowen Qin, Lihan Wang, Binyuan Hui, Bowen Li, Xiangpeng Wei, Binhua Li, Fei Huang, Luo Si, Min Yang, and Yongbin Li. 2022. Sun: Exploring intrinsic uncertainties in text-to-sql parsers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5298–5308.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 32.

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 Recognition Letters*, 88:26–32.

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.

Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for intent classification and out-of-domain detection in slu. 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 2443–2453.

Lei Shu, Yassine Benajiba, Saab Mansour, and Yi Zhang. 2021. Odist: Open world classification via distributionally shifted instances. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3751–3756.

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*, pages 2911–2916.

Ming Tan, Yang Yu, Haoyu Wang, Dakuo Wang, Saloni Potdar, Shiyu Chang, and Mo Yu. 2019. Out-of-domain detection for low-resource text classification tasks. 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 3566–3572, Hong Kong, China. Association for Computational Linguistics.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2021. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*.

Lihan Wang, Bowen Qin, Binyuan Hui, Bowen Li, Min Yang, Bailin Wang, Binhua Li, Jian Sun, Fei Huang, Luo Si, et al. 2022. Proton: Probing schema linking information from pre-trained language models for text-to-sql parsing. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1889–1898.

Wei Wang and Zhi-Hua Zhou. 2017. Theoretical foundation of co-training and disagreement-based algorithms. *arXiv preprint arXiv:1708.04403*.

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020. A large-scale chinese short-text conversation dataset. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 91–103. Springer.

Yida Wang, Yinhe Zheng, Yong Jiang, and Minlie Huang. 2021. Diversifying dialog generation via adaptive label smoothing. 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 3507–3520.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

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*.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020a. 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.

Jin Xu, Zishan Li, Bowen Du, Miaomiao Zhang, and Jing Liu. 2020b. Reluplex made more practical: Leaky relu. In *2020 IEEE Symposium on Computers and communications (ISCC)*, pages 1–7. IEEE.

Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021. Unsupervised out-of-domain detection via pre-trained transformers. 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 1052–1061, Online. Association for Computational Linguistics.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y.S. Lam. 2020a. 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.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. 2020b. 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 (Volume 2: Short Papers)*, pages 870–878, Online. Association for Computational Linguistics.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Hong Xu, and Weiran Xu. 2021b. Adversarial self-supervised learning for out-of-domain detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5631–5639.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert YS 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.

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.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Rongsheng Zhang, Yinhe Zheng, Jianzhi Shao, Xiaoxi Mao, Yadong Xi, and Minlie Huang. 2020. Dialogue distillation: Open-domain dialogue augmentation using unpaired data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3449–3460.

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

Yinhe Zheng, Guanyi Chen, Xin Liu, and Jian Sun. 2022. MMChat: Multi-modal chat dataset on social media. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5778–5786, Marseille, France. European Language Resources Association.

Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020b. A pre-training based personalized dialogue generation model with persona-sparse data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9693–9700.

Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. 2021a. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410.

Hao Zhou, Pei Ke, Zheng Zhang, Yuxian Gu, Yinhe Zheng, Chujie Zheng, Yida Wang, Chen Henry Wu, Hao Sun, Xiaocong Yang, et al. 2021b. Eva: An open-domain chinese dialogue system with large-scale generative pre-training. *arXiv preprint arXiv:2108.01547*.

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021c. Contrastive out-of-distribution detection for pre-trained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022. Knn-contrastive learning for out-of-domain intent classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5129–5141.

A More Examples of Pseudo OOD Samples

This appendix shows more pseudo OOD samples that are generated using existing approaches.

Besides Figure 2, we also visualize pseudo OOD samples produced by Zhan et al. (2021) on the CLINC150 (Larson et al., 2019) dataset (Figure 5) and the StackOverflow (Xu et al., 2015) dataset (Figure 6), when 25% intents are randomly selected as IND intents. Specifically, Zhan et al. (2021) proposes to generate features of pseudo OOD samples by mixing up IND features. Pseudo OOD samples we demonstrate in this analysis are obtained using the code released by Zhan et al. (2021). As shown in Figure 5 and 6, some pseudo OOD samples fall into the cluster of IND intents, and thus we argue it is improper to assign one-hot OOD labels to these samples.

Furthermore, we demonstrate more cases of pseudo OOD samples generated by the method proposed by Shu et al. (2021) on the CLINC150 (Table 5), StackOverflow (Table 6), and Banking (Table 7) datasets. Specifically, these pseudo OOD samples are generated by replacing phrases in IND samples and use a pre-trained language model to filter these samples. As shown in Table 5, 6, and 7, some of the generated pseudo OOD samples carry IND intents since the replaced phrase may be an synonyms of the original phrase.

B More Implementation Details

We use the BERT model (*bert-base-uncased*) provided in the Huggingface's Transformers library (Wolf et al., 2020) to implement f. Following

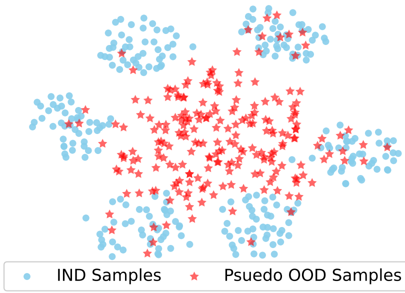

Figure 5: t-SNE visualization of pseudo OOD samples generated by feature mixup (Zhan et al., 2021) on the CLINC150 dataset under the 25% setting. Some pseudo OOD samples coincide with IND samples.

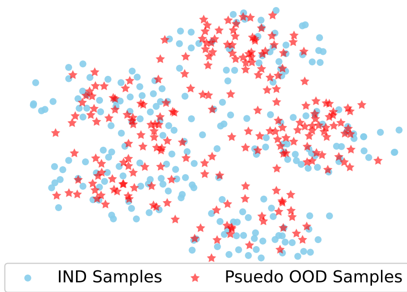

Figure 6: t-SNE visualization of pseudo OOD samples generated by feature mixup (Zhan et al., 2021) on the StackOverflow dataset under the 25% setting. Some pseudo OOD samples coincide with IND samples.

(Zhang et al., 2021), we add an averaging-pooling layer on top of BERT to obtain the representation of each input utterance. The projection head h, classification heads $g1$ and g_2 are implemented as a two-layer MLP with the LeakyReLU activation (Xu et al., 2020b), where the feature dimension is 1024 and the projection dimension is 128. Following (Zhan et al., 2021), We use AdamW (Kingma and Ba, 2014) to fine-tune BERT using a learning rate of 1e-5 and Adam (Wolf et al., 2019) to train the MLP heads using a learning rate of 1e-4 with early stopping. When learning the adaptive decision boundaries (Zhang et al., 2021), the trained model is fixed and we used a learning rate of 0.05. We tried batch size of {100, 200} for IND samples and {100, 500, 600, 800} for OOD samples. All hyper-parameters are tuned according to the classification performance over the IND samples on the validation set. We find that $t = \tau = 0.1$, $\alpha = 0.11$ and $\beta = 0.9$ work well with all datasets. We use a dropout rate 0.6 for the two classification heads. Each result is an average of 10 runs with different random seeds, and each run is stopped when we reach a plateau on the validation loss. ALL

Original Intent	Original Utterance before Distortion	Generated Pseudo OOD Sample
translate	how do you say hi in french	how do you spell hi in french
translate	how do you say please in arabic	how do you say please in Spanish?
transfer	transfer $500 from my checking to my savings	transfer $50 from my checking to my savings
transfer	can you transfer $5 from savings to checking	can you transfer your $5,000 from savings to checking
insurance_change	what do i do for new insurance	what do you do for new insurance
insurance_change	i would like to switch my insurance plan	i had to switch my insurance plan
travel_alert	is it safe to travel to argentina	is it safe to travel to Europe?
travel_alert	tell me about any travel alerts issued for germany	Read more about any travel alerts issued for germany
fun_fact	can you tell me something i don't know about banks	can you tell me something i do n't know about you
fun_fact	can you tell me fun facts about lighthouses	can you tell me fun facts about me?

Table 5: Case study of generated OOD samples with ODIST on the CLINC150 dataset.

Original Intent	Original Utterance before Distortion	Generated Pseudo OOD Sample
wordpress	Multiple loop working, function inside isn't	Multiple looping function inside is n't
wordpress	Plugin to avoid share username in Wordpress	Plugin to share username in Wordpress
visual-studio	Visual Studio Find and Replace Variables	Visual Studio Find and Destroy
visual-studio	Number of Classes in a Visual Studio Solution	Number of Users in a Visual Studio Solution
svn	how to update a file in svn?	how do you open a file in svn?
svn	Revert a whole directory in tortoise svn?	What is a whole directory in tortoise svn?
spring	Problem with Autowiring & No unique bean	Problem with Autowiring & the unique bean
spring	Creating temporyary JMS jms topic in Spring	Creating Your Own Home in Spring
scala	Can Scala survive without corporate backing?	Can Scala get corporate backing?
scala	What is are differences between Int and Integer in Scala?	What is are differences between Int and Int++ in Scala?

Table 6: Case study of generated OOD samples with ODIST on the StackOverflow dataset.

Original Intent	Original Utterance before Distortion	Generated Pseudo OOD Sample
wrong_exchange_rate _for_cash_withdrawal	While abroad I got cash, and a wrong exchange rate was applied.	While abroad I got into an argument with a friend and a wrong exchange rate was applied.
wrong_exchange_rate _for_cash_withdrawal	This past holiday I made a withdraw at the ATM machine, and it seems I've been charged too much.	This morning I made a withdraw at the ATM machine, and it seems I've been charged too much.
wrong_amount_of _cash_received	Where'd the rest of my cash go from the ATM	Where 'd the rest of your cash go from the ATM
wrong_amount_of _cash_received	Why did I get less cash than what I asked in the ATM?	Why did I ask for less cash than what I asked in the ATM?
why_verify_identity	I would like to know why so much identity information is required?	Who would like to know why so much identity information is required?
why_verify_identity	Do i have to verify who I am?	Why do i have to verify who I am?
verify_top_up	How do I verify the top-up?	How do I use the top-up?
verify_top_up	Please tell me how to verify my top up card.	Please check out this post for how to verify my top up card.
verify_my_identity	Let me know what the steps for the identity checks are	We know what the steps for the identity checks are
verify_my_identity	What documents do I need for the identity check?	What documents do we need for the identity check?

Table 7: Case study of generated OOD samples with ODIST on the Banking dataset.

experiments were conducted in the Nvidia Tesla V100-SXM2 GPU with 32G graphical memory. Our model contains 112.06M model parameters.

C More Details about Baselines

Baseline results (MSP, DOC, OpenMax, LMCL, and ADB) are copied from (Zhang et al., 2021). Results of the baseline Outlier are copied from (Zhan et al., 2021). Results of the baseline ODIST are copied from (Shu et al., 2021). For above mentioned baselines, we also re-implement their methods using their release codes. The results reproduced by our experiments match the results reported in their original paper. So we copied the highest reported results of these baselines from previously published papers. Significant tests between our method and all these baselines are carried out based on our implementations. We get the baseline results (SCl and GOT) by running their released codes, and use 100 OOD samples in the validation to determine the thresholds for testing.

D Computational Cost Analysis

We compare the training cost of our method when using one-hot labels or soft labels produced by ASoul. Pseudo OOD samples used in this analysis are generated using the best performing method FM, and we use the CLINC150 dataset for this analysis. As shown in Table 8, ASoul only introduces marginal parameter overhead for the projection head and the classification head. We can also observe that using ASoul only introduces little time overhead compared to the one-hot labeling approach.

Methods	#para.	25%	50%	75%
FM+Onehot	111.36M	7.97s	15.76s	23.40s
FM+Asoul	112.06M	8.57s	16.85s	25.06s

Table 8: Number of parameters (Million) and average training time for each epoch (seconds) on the CLINC150 dataset.

E Effect of Hyper-parameters

We analyzed the most important hyper-parameters of ASoul: temperature τ in graph-based smoothing and dropout rate to the classification heads in co-training. We conduct experiments to show their effects on the CLINC150 dataset under the 25% setting. We use the best performing pseudo OOD sample generation approach (i.e., FM) in this analysis.

Temperature: We set τ to $\{0.1, 0.5, 1, 5, 10, 15, 20\}$ respectively, and demonstrate the performance change. Note that small τ makes distribution over

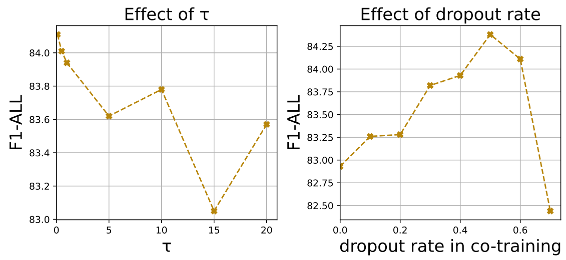

Figure 7: Effect of τ in graph-based smoothing (left) and dropout rate in co-training (right) with FM+ASoul on CLINC150 under the 25% setting.

the embedding graph more shape (concentrating on nearest neighbors), while large τ forms smooth distributions.

Results are shown in Figure 7 (left). With the increase in temperature, the OOD detection performance tends to decrease. $\tau = 0.1$ achieves the highest F1-ALL score of 84.11%. This suggests that a small temperature makes ASoul focus more on neighbors and gain better performance.

Dropout Rate: We compare the performance of dropout rates to the classification heads by adjusting the rate from 0 to 0.7 with an interval of 0.1.

Results are shown in Figure 7 (right). The performance first increases and then decreases as the dropout rate increases. In the begging phase, using a high dropout rate introduces more diversity required by co-training, and thus the OOD detection performance improves. However, using a higher dropout rate introduces much noise to the co-training process, and thus downgrades the OOD detection performance.

F More Evaluation Metrics

We also calculate micro F1-scores over all intents (IND and OOD intents) for our best-performing method FM+ASoul and one of our strongest baselines Outlier on the CLINC150 dataset. As shown in Table 9, FM+ASoul still outperforms the baseline on the micro F1-score.

Methods	25%	50%	75%
Outlier	91.68	88.23	88.46
FM+Asoul	**93.30**	**90.36**	**89.98**

Table 9: Performances of Outlier and FM+Asoul on the CLINC150 dataset under the metric of micro F1-score over all intents (IND and OOD intents) .