# Towards Textual Out-of-Domain Detection without In-Domain Labels

Di Jin, Shuyang Gao, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tür

***Abstract*—In many real-world settings, machine learning models need to identify user inputs that are out-of-domain (OOD) so as to avoid performing wrong actions. This work focuses on a challenging case of OOD detection, where no labels for in-domain data are accessible (e.g., no intent labels for the intent classification task). To this end, we first evaluate different language model based approaches that predict likelihood for a sequence of tokens. Furthermore, we propose a novel representation learning based method by combining unsupervised clustering and contrastive learning so that better data representations for OOD detection can be learned. Through extensive experiments, we demonstrate that this method can significantly outperform likelihood-based methods and can be even competitive to the state-of-the-art supervised approaches with label information.**

***Index Terms*—out-of-domain detection, natural language processing, unsupervised representation learning**

## I. Introduction

DEEP learning models are widely used in many real-life applications, and for many classification problems. At test time, models need to identify examples that differ significantly from the model's training data distribution, i.e., out-of-domain detection. For example, in dialog agents such as Amazon Alexa and Google Home Assistant, detecting unknown or out-of-domain (OOD) intents from user queries is an essential component in order to know when a query falls outside their range of predefined, supported intents [1]. Correctly identifying out-of-scope cases is especially crucial in deployed systems, both to avoid performing the wrong action and also to identify potential future directions for development. However, for state-of-the-art deep learning classifiers, it has been widely observed that their raw probability values are often over-calibrated, i.e., it has high values even for OOD inputs [2], [3]. This necessitates having a specially designed mechanism for OOD detection.

The task of OOD detection has historically been explored in related forms under various names such as outlier detection, anomaly detection, open classification, etc. [3], [4], [5], [6], [7]. Most of previous work relied on the existing class labels (ID labels hereafter) for in-domain multi-class text classification tasks. For example of OOD detection for the intent classification task, existing work implement OOD detection based on a classifier that has been well trained using the intent labels for ID data [1], [8], [9]. In contrast, our work addresses the OOD detection problem in the scenario where no ID labels are available (e.g., no intent labels for the multi-class intent classification task) and thus most existing solutions are not directly applicable. One real-world use case for this scenario could be the zero-shot intent classification task, which commonly happens in practice especially for early prototyping or extending a developed system to a new domain. Existing work have shown successful results without using any ID labels [10], [11], however, these studies assume that the data at test time would contain only in-domain examples. Our work can augment these systems with new capability to detect out OOD samples for real-life user data while maintaining the zero-shot capability.

Current available methods for tackling this *OOD detection without ID labels* setting can be categorized into two threads: language model (LM) likelihood based methods [12], [13], and representation learning based one-class classification method [14], [15]. For the first thread, [16], [17] find that likelihood is poor at separating out OOD examples; in some cases even assigning them higher likelihood than the ID test split. [12] made similar observations for detecting OOD DNA sequences and thus proposed to "correct" the original likelihood with one from a "background" model trained on noisy inputs, termed as likelihood ratio (LR). [13] explored the use of LR method on NLP tasks and reported good performance. For the second thread, we can treat all of the ID data as one-class data and those OOD data as other classes, forming a one-class classification problem. [15] proposed to first learn a good representation model via contrastive learning and then learn a density estimator on the obtained representations, which has shown state-of-the-art performance for one-class classification.

In this work, we propose novel methods for both of the above-mentioned threads for NLP tasks. Specifically, for the first thread of LM likelihood based methods, we thoroughly evaluate several likelihood based approaches, including that from [13]. We demonstrate that the background model trained on noisy inputs is not effective for most cases, and length normalization for LM likelihood is often needed for proper score evaluation. For the other direction, we find that directly applying representation learning based one-class classification methods to our problem cannot yield satisfying performance, since ID data indeed consists of multiple classes that we do not have access to. To solve this issue, we propose to perform both unsupervised clustering and contrastive learning when we learn representations of ID data, which can help produce sharper boundaries between ID and OOD samples. To the best of our knowledge, we are the first to apply the representation learning based one-class classification method to OOD detection in NLP tasks and significantly improve it so that it can rival likelihood based methods. In summary, our contributions are listed below:

All authors are with Amazon Alexa AI, Sunnyvale, USA, 94089, USA.
Manuscript received April 19, 2005; revised August 26, 2015.

- We evaluate different LR methods using LM likelihood and augmenting factors including length normalization and background LMs. We demonstrate that a pre-trained background LM without any fine-tuning is even better for OOD detection than currently published background LMs trained on noisy texts.
- We for the first time adopt representation learning based one-class classification method for textual OOD detection and enhance the representation learning process by introducing contrastive learning based unsupervised clustering, which plays a key role in improving detection performance for this line of methods.
- We have conducted a thorough comparison between the LM likelihood based and representation learning based methods, and demonstrate that with proper unsupervised representation learning methods, the latter line of approaches can outperform the other counterpart.
- Overall, our proposed methods for textual OOD detection without ID labels have achieved very impressive performance, even close to those achieved by supervised methods that use ID labels for training for three out of four datasets.[1]

## II. Related Works

### A. OOD Detection

OOD detection has a long history [4]. Recent studies on this problem include [3] that proposed benchmark datasets on vision problems, and [18] where OOD data were curated and used by training models to increase entropy on OOD samples. However, in real-life NLP applications, gathering and maintaining OOD data is complicated by the lexical and stylistic variation of natural language [19]. For this reason, methods without OOD supervision gain more attention. This is referred as **unsupervised** OOD detection, which is the focus of this work.

All OOD detection methods for unsupervised detection rely on producing a score that can be compared with a threshold to differentiate between ID and OOD samples. One major line of methods utilize classifiers trained on the ID data to calculate the **probability** of the test instance and use it as comparison score. The maximum softmax probability is recognized as a strong baseline [3]. [20] found that increasing the softmax temperature $\tau$ makes the resulting probability more discriminative for OOD Detection. [6] and [21] utilized ID inputs and unlabeled data to generate pseudo-OOD utterances around the decision boundaries with a generative adversarial network (GAN) so as to calibrate the output probabilities. Another important line of methods use **distance estimation** as the OOD score: [7] proposed using Mahalanobis distances to per-class Gaussians in the intermediate representation learned by the classifier. Specifically, a Gaussian distribution is fit for each training class from all training points in that class. Following this work, [22] analyzed the effectiveness of Mahalanobis distance and proposed several variants. [23] and [8] applied it to NLP tasks and reported strong performance.

[1]The source code will be released upon publication.

All of the above previous work assumed the availability of well-trained classifiers on ID data with annotated ID labels. However, in some cases, such annotated labels for classification may not be available, posing a great challenge to OOD detection. Our work will focus on addressing this special case of **OOD detection without ID labels**. It has been rarely explored previously. The only available ready-to-use method is the likelihood ratio method, where two deep generative models are trained on normal and noisy inputs, respectively, and then the difference of likelihood produced by these two models is used as the OOD score [12], [13], [8]. In this study, we compare that work with other ways of building the background model and evaluate the effect of length normalization for likelihood scores. We find that a generic pre-trained language model without domain adaptation is a better choice for the background model.

### B. One-class Classification

From another perspective, we can treat all ID data as one class and all OOD data as open classes, then we can adopt those one-class classification methods for our problem. Along this path, recent work has focused on utilizing self-supervised learning methods such as rotation learning [24] and contrastive learning [15], [25] for representation learning and then adopting a density estimator to differentiate the target class from others. While we adopt this framework for textual OOD detection for the first time, we find its performance is unsatisfying. The reason is that in our problem setting, ID data are not strictly one class but indeed contain multiple hidden classes, which is not accommodated by existing one-class classification methods. Therefore, we propose an unsupervised clustering method augmented by contrastive learning [26], [27] as the self-supervised method for representation learning, and show this leads to significant improvement on OOD detection.

## III. Methods

Overall for OOD detection given an input text $\mathbf{x}$, we need to produce a score $s$ for it, which is used to be compared with a threshold $\eta$ to determine whether this is an ID sample or an OOD one. Such a score can be a likelihood score produced by a language model or can be a density score produced by a density estimator. We therefore propose the LM Likelihood based method for the former kind of likelihood score in Section III-A, while proposing the representation learning based method for the density score in Section III-B.

### A. LM Likelihood Based

In this section, we introduce various methods based on language models (LM). The basic idea using LM for OOD is building a LM using the in-domain data and then checking how likely a test instance is generated by that LM. The probability of an instance can be estimated using traditional n-gram based LMs. In this work, we adopt recent neural networks as LMs.

We evaluate the following approaches.

- LN (likelihood normalized).

$$LN(\mathbf{x}) = \Big[\prod_{i=1}^{|S|} P_{M_{ID}}(x_i|\mathbf{x}_{<i})\Big]^{1/|S|} \quad (1)$$

where $M_{ID}$ represents the LM trained on the in-domain data, $|S|$ is the length of instance $\mathbf{x}$, $x_i$ denotes the $i^{th}$ token in $\mathbf{x}$, and $\mathbf{x}_{<i}$ denotes all tokens before $x_i$ in $\mathbf{x}$. This is the likelihood with length normalization using in-domain LM, and is equivalent to the widely used LM perplexity metric.

- LR (likelihood ratio).

$$LR(\mathbf{x}) = \frac{P_{M_{ID}}(\mathbf{x})}{P_{M_B}(\mathbf{x})} = \frac{\prod_{i=1}^{|S|} P_{M_{ID}}(x_i|\mathbf{x}_{<i})}{\prod_{i=1}^{|S|} P_{M_B}(x_i|\mathbf{x}_{<i})}, \quad (2)$$

where $P_{M_{ID}}(\mathbf{x})$ and $P_{M_B}(\mathbf{x})$ denote the probability of $\mathbf{x}$ produced by the in-domain LM ($M_{ID}$) and a background LM ($M_B$), respectively.
There are different ways to obtain the background LM. In this study, we use a generic pre-trained language model such as GPT-2 without any fine-tuning as the background LM.
In the literature, [13] also proposed using LR for OOD in NLP tasks, where they trained the in-domain model $M_{ID}$ on the original training corpus $X = \{\mathbf{x}^1, ..., \mathbf{x}^N\}$ via a language modeling task, and the background model $M_B$ on a noisy version of training samples $\hat{X} = \{\hat{\mathbf{x}}^1, ..., \hat{\mathbf{x}}^N\}$ generated with random word substitutions. Specifically, with probability $p_{noise}$, each word $x_i$ in sentence $\mathbf{x}$ is randomly selected and substituted with word $x_i'$ that is sampled from the vocabulary $V$ with distribution $N(x_i') = \frac{\sqrt{f(x_i')}}{\sum_{x \in V}\sqrt{f(x)}}$, where $f(x)$ is the word frequency. We adopt their approach in this work for comparison. We call this approach $LR_{ws}$, and will show in the experiments later that such a way to create background corpus is often not appropriate.
The difference between LR and LN is in that no background models are used in LN, and thus LN is more computationally efficient. Note that LR methods have been widely used in many tasks. For example, in speaker verification or recognition, LR is calculated using the speaker specific model with respect to the universal background model [28]. LR was also used for vision and DNA sequencing tasks [12].

- NLR (length normalized likelihood ratio). Considering that length is an important factor in LM probabilities, we also evaluate using the ratio of the normalized likelihood for OOD. Similar to LR above, we will use an off-the-shelf pre-trained language model as the background LM. NLR is defined as below:

$$NLR(\mathbf{x}) = \frac{\left[\prod_{i=1}^{|S|} P_{M_{ID}}(x_i|\mathbf{x}_{<i})\right]^{1/|S|}}{\left[\prod_{i=1}^{|S|} P_{M_B}(x_i|\mathbf{x}_{<i})\right]^{1/|S|}}. \quad (3)$$

### B. Representation Learning Based

In this section, we would like to propose a representation learning based method for OOD detection without any ID labels. More specifically, the representation learning is implemented via a classic unsupervised method: unsupervised clustering. Combined with a recent popular technique, contrastive learning, it can significantly boost the OOD detection performance.

The training process of our method is mainly composed of two steps: adaptive representation learning and density estimation.

- **Adaptive representation learning:** In this step, we learn the ID data distribution with a pre-trained sentence encoder $\psi$ by optimizing the following overall objective:

$$\mathcal{L} = \mathcal{L}_{cluster} + \gamma\mathcal{L}_{CL}, \quad (4)$$

where $\mathcal{L}_{cluster}$ denotes the unsupervised clustering objective and $\mathcal{L}_{CL}$ denotes the contrastive learning objective, both of which are described in more details below, and $\gamma$ is tuned on the validation set.
- **Density estimation:** In this step, we feed each ID sentence $\mathbf{x}_i^{ID}$ into the encoder $\psi$ and obtain the mean vector of all the tokens' hidden states from the last layer, which is used as its representation vector: $\psi(\mathbf{x}_i^{ID})$. Then we learn a density estimator $D$ over these vectors, such as One-class Support Vector Machine (OC-SVM), Kernel Density Estimation (KDE), and Gaussian Mixture Model (GMM). We choose GMM in our work since we find it consistently outperforms other choices in experiments.

The above-mentioned training process would lead to an encoder $\psi$ and a density estimator $D$ with well-trained parameters. For inference, given a test sample $\mathbf{x}$, we first encode it with $\psi$ and then use the density estimator $D$ to produce a density score $D(\psi(\mathbf{x}))$. If this score is above a pre-set threshold $\eta$, this sample is considered as an ID sample, otherwise as an OOD sample.

In the following, we describe the representation learning in detail.

*1) Clustering:* Our ID data may contain multiple classes, but we do not have such annotations. Therefore, we would like to adopt unsupervised clustering to learn this implicit prior. Suppose our ID data consists of $K$ semantic categories, and each category is characterized by its centroid in the representation space, denoted as $\mu_k, k \in \{1, ..., K\}$. Here $\mu_k$ is initialized via K-means clustering and then iteratively refined during the training phase. Note that since we do not know how many clusters there are for the ID data, $K$ is treated as a hyper-parameter and tuned using the validation set. Overall, the objective for unsupervised clustering is to push the cluster assignment probability $q_i$ for input text $x_i$ towards the target distribution $p_i$, which is achieved by optimizing the KL divergence between them:

$$\ell_i^C = KL(p_i||q_i) = \sum_{k=1}^{K} p_{ik} \log \frac{p_{ik}}{q_{ik}},$$

where $q_{ik}$ and $p_{ik}$ are the predicted and target probability of assigning $x_i$ to the $k^{th}$ cluster, respectively. The overall clustering objective is then defined as:

$$\mathcal{L}_{cluster} = \frac{1}{M}\sum_{i=1}^{M} \ell_i^C, \quad (5)$$

where $M$ is the batch size.

Following [29], we use the Student's t-distribution to compute $q_{ik}$ as below:

$$q_{ik} = \frac{(1+ \parallel e_i - \mu_k \parallel_2^2 /\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'=1}^{K}(1+ \parallel e_i - \mu'_k \parallel_2^2 /\alpha)^{-\frac{\alpha+1}{2}}},$$

where $e_i = \psi(x_i)$, and $\alpha$ denotes the degree of freedom of the Student's t-distribution, which is set as 1 in this work.

Since we do not have the ground truth of $p_{ik}$, we approximate it following [30]:

$$p_{ik} = \frac{q_{ik}^2/f_k}{\sum_{k'=1}^{K} q_{ik'}^2/f_{k'}},$$

where $f_k = \sum_{i=1}^{M} q_{ik}$. This target distribution first sharpens the soft-assignment probability $q_{ik}$ by raising it to the second power, and then normalizes it by the associated cluster frequency. By doing so, we encourage learning from high confidence cluster assignments and simultaneously combating the bias caused by imbalanced clusters.

*2) Contrastive Learning:* Following [26], we add a contrastive learning training objective [25] while performing clustering, which can help stabilize and improve the clustering process. For a randomly sampled mini-batch $\mathcal{B} = \{x_{i^0}\}_{i^0=1}^{M}$ (all come from ID samples), we randomly generate an augmentation $x_{i^1}$ for each data instance $x_{i^0}$ so that $x_{i^0}$ and $x_{i^1}$ form a pair of positive instances, yielding an augmented mini-batch $\mathcal{B}^a$ of size $2M$. Following [27], to obtain the data augmentation, we pass the same input sentence to the pre-trained sentence encoder twice and obtain two embeddings as "positive pairs", by applying independently sampled dropout masks (current prevalent pre-trained encoders are all based on transformer architecture and contain dropout masks in each layer). Although strikingly simple, this approach has been found to outperform many other complex text augmentation methods, such as contextual word replacement and back-translation [27]. To achieve contrastive learning, for $x_{i^0}$ we try to bring it close to its positive counterpart $x_{i^1}$ while moving it far away from other instances in the mini-batch $\mathcal{B}^a$, which is implemented by minimizing:

$$\ell_{i^0}^{CL} = -\log \frac{\exp(sim(z_{i^0}, z_{i^1})/\tau)}{\sum_{j=1}^{2M} \mathbb{1}_{j \neq i^0} \cdot \exp(sim(z_{i^0}, z_j)/\tau)},$$

where $z_j = g(\psi(x_j))$ and $g$ is implemented by a two-layers full connected network in this work; $\tau$ denotes the temperature parameter which we set as 0.5; $sim(z_i, z_j) = z_i^T z_j / ||z_i||_2 ||z_j||_2$ following [25]. Then the overall contrastive learning objective is defined as:

$$\mathcal{L}_{CL} = \frac{1}{2M} \sum_{i=1}^{2M} \ell_i^{CL}. \quad (6)$$

## IV. Experiments

### A. Datasets

We have experimented with four text classification datasets: two on intent classification for dialogue systems, one on question topic classification, and one on search snippets topic classification.

*a) SNIPS:* consists of user utterances for 7 intent classes such as GetWeather, RateBook, etc. [31]. Since this dataset itself does not include OOD utterances, we follow the procedure described in [32] to synthetically create OOD examples. Intent classes covering at least 75% of the training points in combination are retained as ID. Examples from the remaining classes are treated as OOD and removed from the training set. In the validation and test sets, examples from these classes are relabelled to the single class label OOD. Since multiple ID-OOD splits of the classes satisfying these ratios are possible, our results are averaged across 5 randomly chosen splits.

*b) ROSTD:* starts from the English part of multilingual dialog dataset released by [33] as ID utterances and later gets extended by [13] with OOD utterances.

*c) Stackoverflow:* is a subset of the challenge data published by Kaggle,[2] where question titles associated with 20 different categories are selected by [34]. We follow the same way as SNIPS to create OOD samples. We report average results across 5 randomly chosen splits.

*d) Searchsnippets:* is extracted from web search snippets, which contains search snippets associated with 8 different topics [35]. We randomly selected 2 out of 8 classes as OOD classes while using the remaining classes as ID classes. Again we report average results across 5 randomly chosen splits.

Table I summarizes the statistics of these four datasets. To be noted, the labels in these datasets are only used for splitting ID and OOD samples and not used for training the OOD detector.

TABLE I
DATASET STATISTICS. EXCEPT ROSTD, ALL NUMBERS ARE AVERAGED ACROSS 5 CHOSEN SPLITS.

| Statistic | ROSTD | SNIPS | Stackoverflow | Searchsnippets |
|---|---|---|---|---|
| Train-ID | 30,521 | 9,332 | 10,020 | 9,163 |
| Valid-ID | 4,181 | 500 | 428 | 1,238 |
| Valid-OOD | 1,500 | 200 | 229 | 1,292 |
| Test-ID | 8,621 | 506 | 418 | 1,237 |
| Test-OOD | 3,090 | 193 | 235 | 1,293 |

### B. Evaluation Metrics

Following [3], we use the following metrics to measure OOD detection performance:

*a)* $\mathbf{FPR@95\%TPR}$*:* corresponds to False Positive Ratio with the decision threshold being set to $\theta = sup\{\tilde{\theta} \in \mathcal{R} | TPR(\tilde{\theta}) \leq 95\%\}$, where $TPR$ is the True Positive Ratio. Note here the OOD class is the positive class.

*b)* $\mathbf{AUROC}$*:* measures the area under the Receiver Operating Characteristic, also known as the ROC curve. Note that this curve is for the OOD class.

*c)* $\mathbf{AUPR_{OOD}}$*:* measures the area under Precision-Recall Curve, taking OOD as the positive class. It is more suitable for highly imbalanced data in comparison to $AUROC$.

[2]https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow

TABLE II
OOD DETECTION PERFORMANCE OF LM LIKELIHOOD BASED METHODS ON ALL FOUR DATASETS. SINCE WE HAVE 5 RANDOM SPLITS FOR SNIPS, STACKOVERFLOW, AND SEARCHSNIPPETS DATASETS, WE REPORT THE AVERAGE AND STANDARD DEVIATION RESULTS.

| Dataset | Likelihood | Model | $\mathbf{AUROC}(\%)\uparrow$ | $\mathbf{AUPR_{OOD}}(\%)\uparrow$ | $\mathbf{FPR@95\%TPR}(\%)\downarrow$ |
|---|---|---|---|---|---|
| ROSTD | $LR_{ws}$ | LSTM | 96.81 | 93.97 | 13.52 |
| | LN | LSTM | 98.82 | 96.91 | 5.13 |
| | LN | GPT-2 Medium | 98.89 | 97.19 | 4.74 |
| | LR | GPT-2 Medium | 99.23 | 97.47 | 3.43 |
| | NLR | GPT-2 Medium | **99.31** | **98.23** | **2.95** |
| SNIPS | $LR_{ws}$ | LSTM | $93.38 \pm 0.82$ | $85.55 \pm 2.44$ | $28.25 \pm 2.63$ |
| | LN | LSTM | $92.37 \pm 0.61$ | $79.21 \pm 2.23$ | $25.79 \pm 2.09$ |
| | LN | GPT-2 Medium | $93.68 \pm 2.38$ | $82.19 \pm 7.79$ | $25.37 \pm 9.41$ |
| | LR | GPT-2 Medium | $\mathbf{95.31} \pm 2.22$ | $\mathbf{88.51} \pm 5.49$ | $\mathbf{21.76} \pm 8.91$ |
| | NLR | GPT-2 Medium | $94.42 \pm 2.35$ | $86.53 \pm 6.92$ | $25.70 \pm 10.77$ |
| Stackoverflow | $LR_{ws}$ | LSTM | $69.96 \pm 1.03$ | $51.86 \pm 1.90$ | $77.68 \pm 0.46$ |
| | LN | LSTM | $72.43 \pm 0.64$ | $57.41 \pm 1.42$ | $76.86 \pm 1.12$ |
| | LN | GPT-2 Medium | $77.76 \pm 1.22$ | $60.78 \pm 3.21$ | $69.12 \pm 3.47$ |
| | LR | GPT-2 Medium | $\mathbf{79.77} \pm 1.57$ | $60.97 \pm 2.31$ | $\mathbf{61.63} \pm 6.44$ |
| | NLR | GPT-2 Medium | $78.50 \pm 1.79$ | $\mathbf{61.94} \pm 1.88$ | $65.29 \pm 3.70$ |
| Searchsnippets | $LR_{ws}$ | LSTM | $82.11 \pm 0.75$ | $86.65 \pm 0.47$ | $61.78 \pm 1.16$ |
| | LN | LSTM | $81.75 \pm 2.57$ | $85.96 \pm 2.69$ | $61.80 \pm 1.16$ |
| | LN | GPT-2 Medium | $82.40 \pm 1.51$ | $87.25 \pm 1.67$ | $60.12 \pm 3.17$ |
| | LR | GPT-2 Medium | $80.30 \pm 3.25$ | $84.13 \pm 2.27$ | $60.67 \pm 7.98$ |
| | NLR | GPT-2 Medium | $\mathbf{82.68} \pm 4.09$ | $\mathbf{87.44} \pm 3.07$ | $\mathbf{59.22} \pm 9.12$ |

### C. Experimental Settings

*1) LM Likelihood Based Methods:* For $LR_{ws}$, we follow [13] and used LSTM as the LM. We set $p_{noise}$ to 0.5 in $LR_{ws}$, which has been found to be optimal by [13]. For $LN$, we tried both LSTM and GPT2-Medium [36] for the LM. For LSTM, we vary the hidden state dimension among $\{64, 128, 256\}$ and choose the one with highest validation set OOD detection performance. To learn the in-domain LM $M_{ID}$, we train both LSTM and GPT2 on the used datasets via optimizing the unsupervised auto-regressive language modeling objective, where LSTM is trained from scratch while GPT2 is initialized with pre-trained parameters. When GPT2 is used as the background LM $M_B$, it is initialized with pre-trained parameters.

*2) Representation Learning Based Methods:* The sentence encoder $\psi$ we used is DistilBERT-Base-NLI (denoted also as DistilBERT in the rest of this document), which is obtained by fine-tuning DistilBERT-Base on MultiNLI and SNLI datasets [37].[3] By tuning on the validation set, we set the number of components for GMM as 1 and set $\gamma$ as 1.0, 0.1, 1.0, and 4.0 for ROSTD, SNIPS, Stackoverflow, and Searchsnippets, respectively.

### D. Baselines

We compare our representation learning based method with the following baselines:

*a) Encoder w/o Fine-tuning:* We directly use a pre-trained encoder $\psi$ as an off-the-shelf model without any fine-tuning. Specifically, we used DistilBERT as well as the RoBERTa-Large-NLI-SimCSE (denoted also as RoBERTa later) that is obtained by fine-tuning RoBERTa-Large on NLI datasets via contrastive learning and is the state-of-the-art encoder for sentence embedding [27].[4] We report the best result between these two encoders.

*b) Fine-tuned Encoder via MLM:* We fine-tune DistilBERT and RoBERTa on ID data via masked language modeling (MLM) [38] to obtain domain data adapted $\psi$ and report the best result.

*c) DistilBERT/RoBERTa Maha:* We re-implemented a supervised method from [8] by training a DistilBERT and a RoBERTa model on ID data with those ID labels and then performing OOD detection via Mahalanobis distance, which is the state-of-the-art method for textual OOD detection.

*d) $-\mathcal{L}_{cluster}$ & $-\mathcal{L}_{CL}$:* Either unsupervised clustering by optimizing Equation 5 or contrastive learning by optimizing Equation 6 can act as strong baselines since they can both individually serve as approaches for unsupervised representation learning. We also compare with these two baselines in the ablation study.

## V. RESULTS

### A. LM Likelihood Based Methods

We first present results for the LM likelihood based methods in Table II. We can see that overall LR and NLR with GPT-2 as the background LM achieve the best results for all the datasets. The benefit of length normalization in NLR compared with LR is not consistent across the four datasets, which we hypothesize is because the likelihood is already normalized using the background LM in the LR method. In contrast, for LN, since it only uses the LM likelihood from the in-domain model, the likelihood score needs to be properly normalized by the sequence length (given how the chain rule is used in probability calculation). Such a simple normalization

[3]https://github.com/UKPLab/sentence-transformers

[4]https://github.com/princeton-nlp/SimCSE

TABLE III
COMPARISON OF OOD DETECTION PERFORMANCE FOR THE REPRESENTATION LEARNING BASED METHODS. THE BEST RESULTS ACHIEVED BY LIKELIHOOD BASED METHODS FROM TABLE II ARE INCLUDED AS ONE BASELINE (REFERRED AS LIKELIHOOD-BEST). THE BOLD FONT HIGHLIGHTS THE BEST RESULTS ACHIEVED AMONG METHODS WITHOUT USING ID LABELS. † DENOTES THAT THE BEST RESULT IS ACHIEVED BY THE ROBERTA-LARGE-NLI-SIMCSE ENCODER WHILE ‡ REPRESENTS IT IS FROM DISTILBERT-BASE-NLI. * DENOTES THAT THE RESULT HAS SHOWN STATISTICAL SIGNIFICANCE COMPARED WITH BEST BASELINE WITH $p < 0.05$. "CL" DENOTES "CONTRASTIVE LEARNING".

| Dataset | ID Labels | Model | AUROC(%) ↑ | $\mathbf{AUPR_{OOD}}$(%) ↑ | FPR@95%TPR(%) ↓ |
|---|---|---|---|---|---|
| ROSTD | NO | Likelihood-Best | 99.31 | 98.23 | 2.95 |
| | | Encoder w/o Fine-tuning† | **99.77** | **99.46** | **0.71** |
| | | Fine-tuned Encoder via MLM† | 99.29 | 98.11 | 3.00 |
| | | Ours | 99.73 | 99.30 | 0.73 |
| | | $-\mathcal{L}_{cluster}$ | 97.78 | 93.91 | 9.58 |
| | | $-\mathcal{L}_{CL}$ | 98.88 | 96.45 | 4.87 |
| | YES | DistilBERT Maha | 99.61 | 98.85 | 1.60 |
| | | RoBERTa Maha | 99.80 | 99.50 | 0.50 |
| SNIPS | NO | Likelihood-Best | $95.31 \pm 2.22$ | $88.51 \pm 5.49$ | $21.76 \pm 8.91$ |
| | | Encoder w/o Fine-tuning† | $94.09 \pm 3.07$ | $82.31 \pm 10.04$ | $17.78 \pm 7.08$ |
| | | Fine-tuned Encoder via MLM‡ | $94.76 \pm 3.31$ | $86.53 \pm 8.07$ | $21.54 \pm 15.03$ |
| | | Ours | $\mathbf{97.56}^* \pm 1.01$ | $\mathbf{92.80}^* \pm 3.19$ | $\mathbf{8.53}^* \pm 4.42$ |
| | | $-\mathcal{L}_{cluster}$ | $84.54 \pm 8.95$ | $64.02 \pm 18.70$ | $43.57 \pm 19.08$ |
| | | $-\mathcal{L}_{CL}$ | $97.17 \pm 1.37$ | $92.22 \pm 3.97$ | $15.75 \pm 8.93$ |
| | YES | DistilBERT Maha | $97.93 \pm 1.37$ | $95.29 \pm 3.03$ | $8.65 \pm 6.42$ |
| | | RoBERTa Maha | $98.30 \pm 0.98$ | $95.10 \pm 3.80$ | $7.06 \pm 3.51$ |
| Stackoverflow | NO | Likelihood-Best | $79.77 \pm 1.57$ | $60.97 \pm 2.31$ | $61.63 \pm 6.44$ |
| | | Encoder w/o Fine-tuning‡ | $73.07 \pm 3.77$ | $55.69 \pm 3.40$ | $76.54 \pm 1.62$ |
| | | Fine-tuned Encoder via MLM‡ | $78.59 \pm 1.68$ | $62.18 \pm 1.63$ | $64.13 \pm 5.04$ |
| | | Ours | $\mathbf{83.65}^* \pm 0.13$ | $\mathbf{62.22} \pm 2.85$ | $\mathbf{36.82}^* \pm 1.84$ |
| | | $-\mathcal{L}_{cluster}$ | $70.99 \pm 1.99$ | $53.76 \pm 3.00$ | $78.43 \pm 3.32$ |
| | | $-\mathcal{L}_{CL}$ | $50.79 \pm 13.13$ | $37.99 \pm 13.54$ | $90.80 \pm 5.24$ |
| | YES | DistilBERT Maha | $92.00 \pm 0.96$ | $78.35 \pm 3.24$ | $19.99 \pm 3.02$ |
| | | RoBERTa Maha | $91.98 \pm 1.34$ | $76.55 \pm 5.62$ | $19.61 \pm 2.51$ |
| Searchsnippets | NO | Likelihood-Best | $82.68 \pm 4.09$ | $87.44 \pm 3.07$ | $59.22 \pm 9.12$ |
| | | Encoder w/o Fine-tuning† | $82.93 \pm 5.05$ | $88.22 \pm 3.78$ | $61.80 \pm 11.02$ |
| | | Fine-tuned Encoder via MLM† | $84.32 \pm 1.96$ | $89.27 \pm 2.15$ | $62.54 \pm 4.24$ |
| | | Ours | $\mathbf{94.70}^* \pm 2.29$ | $\mathbf{96.16}^* \pm 1.92$ | $\mathbf{22.43}^* \pm 7.20$ |
| | | $-\mathcal{L}_{cluster}$ | $88.09 \pm 6.63$ | $90.91 \pm 5.73$ | $41.09 \pm 17.76$ |
| | | $-\mathcal{L}_{CL}$ | $78.95 \pm 4.93$ | $82.01 \pm 4.22$ | $65.06 \pm 11.53$ |
| | YES | DistilBERT Maha | $95.09 \pm 2.43$ | $96.88 \pm 1.56$ | $25.13 \pm 14.00$ |
| | | RoBERTa Maha | $97.26 \pm 1.05$ | $98.38 \pm 0.64$ | $14.61 \pm 7.00$ |

strategy by using the sequence length can avoid using a second LM, and thus it is computationally more efficient, while its performance is actually comparable to both LR and NLR across all datasets. The comparison of LSTM and GPT2 as the in-domain LM for LN shows that GPT2 is a more powerful LM and has better quality in its likelihood estimation.

Finally, $LR_{ws}$ is not competitive in most cases compared to LN, both of which use LSTM as their LMs. It indicates that substituting randomly selected words with other words randomly drawn from the vocabulary to create noisy texts for training the background LM is not effective. Intuitively, training a LM on perturbed texts that are totally not grammatical can learn nothing in linguistics but the sentence length, considering that the likelihood of the whole sequence is the product of all token probabilities and each token probability would be biased to be uniformly distributed over the vocabulary when learned on randomly perturbed text. This can help explain why $LR_{ws}$ cannot even win over LN in most cases since LN can also remove the influence of sentence length via normalization. We provide a more detailed analysis over $LR_{ws}$ to explain the disadvantages of this method in the Section VI-A.

The large performance variation of the same methods across different datasets reveals the diverse difficulty levels of OOD detection across datasets, and the difficulty level depends on both the language variation within ID data and the overlap of data distributions between ID and OOD data. For example, in the ROSTD dataset, models achieve almost perfect OOD detection performance. This is because its language variation in the samples for each intent is very limited (e.g., many utterances for the same intent only differ in one or two tokens), and its OOD samples differ significantly from the ID ones (e.g., consisting of subjective and emotional utterances), leading to lesser overlap between ID and OOD data than other datasets.

### B. Representation Learning Based Methods

Table III summarizes the main results of OOD detection for the representation learning based methods. We have included the best results achieved by those likelihood based methods

from Table II as one baseline for comparison. From this table, we have the following observations:

- Simply applying a generic pre-trained encoder without any fine-tuning on the ID data to obtain data representations and then performing density estimation can already achieve good performance (refer to Encoder w/o Fine-tuning), sometimes comparable to the best likelihood method (refer to Likelihood-Best).
- Adapting the pre-trained encoder on the ID data via MLM, i.e., fine-tuned encoder via MLM, shows some improvement for most datasets and metrics, e.g., on SNIPS, Stackoverflow, and Searchsnippets datasets for the $AUROC$ and $AUPR_{OOD}$ metrics.
- Our proposed method for adapting the encoder via clustering and contrastive learning can significantly boost the OOD detection performance consistently for all datasets, outperforming all baselines.
- Representation learning based methods can outperform likelihood based methods significantly on all datasets. For example of the Searchsnippets dataset, the performance gap between these two lines of methods is over 10% for all three metrics.
- Our method can be comparable to the state-of-the-art method that intensively utilizes the ID labels for supervised training (i.e., DistilBERT/RoBERTa Maha) for ROSTD, SNIPS, and Searchsnippets datasets.

The key to the success of OOD detection is to learn precise boundaries of ID data distribution, which are used to differentiate ID and OOD samples. Likelihood based methods via language modeling treat all in-domain data points equally and learn their distribution boundaries without explicitly exploring their internal distribution patterns. However, the representation based method can better make use of the fact that there exist some clusters for ID samples but we just do not have labels for them. The learned boundaries by taking into account internal clusters among ID samples can be more tight that those by treating all ID data as a single cluster, which could be the main reason why representation learning based methods significantly outperform likelihood based methods.

It is worth pointing out that our method is based on a small-size encoder, i.e., DistilBERT-Base (around 66M parameters), however, those baselines including Encoder w/o Fine-tuning, Finetuned Encoder, and RoBERTa Maha have adopted an encoder of much larger size, i.e., RoBERTa-Large with around 355M parameters. Moreover, the RoBERTa-Large encoder has been pre-trained on around 10 times larger corpora compared with DistilBERT-Base, thus achieving much better performance on various kinds of language understanding tasks, e.g., the average score of 88.5 achieved by RoBERTa-Large vs 77.0 achieved by DistilBERT-Base on the development sets of GLUE benchmark [39], [40]. Although our method is equipped with an encoder with much smaller model size (about 1/5) and inferior down-stream tasks performance, it can still beat those baselines using larger models when no ID labels are used, and be on par with the baseline that does use ID labels. This comparison demonstrates that our method can achieve both high OOD detection performance and parameter & computation efficiency, which fulfills the two consideratas for real-world deployment in industry applications.

Among all datasets, it can be observed that the advantage of our method over baselines for the Stackoverflow dataset is not as large as other datasets. This is mainly because this dataset is more challenging and the unsupervised clustering performance is not good on it. On one hand, this task aims to classify a short question title (usually around 10 tokens) from the Stackoverflow website into one technological topic, e.g., excel, oracle, bash, apache, etc., which is challenging in its nature. On the other hand, the best clustering algorithm in [34] can only achieve 51.14% of accuracy and the SOTA OOD model with supervised ID labels in our Table III (i.e., DistilBERT Maha) can only obtain 78.35% of $AUPR_{OOD}$, which further validates its difficulty level.

## VI. Discussions

### A. Analysis of $LR_{ws}$

In order to further analyze the method $LR_{ws}$ reported in [13], we first plot histograms of $\log P_{M_{ID}}(\mathbf{x})$, $\log P_{M_B}(\mathbf{x})$, $\log LR_{ws}(\mathbf{x})$, and $\log LN(\mathbf{x})$ for the ROSTD dataset using LSTM as the LM, as shown in Figure 1. As can be seen from Figure 1B, the likelihood $P_{M_B}(\mathbf{x})$ computed by the background model $M_B$ itself, which is trained on text with noise introduced, cannot differentiate between ID and OOD samples. The overlap between histograms of ID and OOD samples of $P_{M_B}(\mathbf{x})$ (Figure 1B) is even larger than that of $P_{M_{ID}}(\mathbf{x})$ (Figure 1A). However, their ratio, $LR_{ws}$, presents less overlap between ID and OOD samples (Figure 1C). To explore the reason why subtracting the likelihood of background model can help differentiate ID and OOD samples, we further calculate the Pearson correlation between the likelihoods and sentence length for all four datasets, which is summarized in Table IV. From this table, we find that $P_{M_B}(\mathbf{x})$ has a strong correlation with sentence length for all four datasets, indicating that it carries abundant information about sentence length. In contrast, $LR_{ws}$ presents much less correlation with the sentence length, which also shows less overlap between histograms of ID and OOD samples as shown in Figure 1C. This finding leads to our hypothesis that the key to the success of $LR_{ws}$ could be that the likelihood of the background model can help remove the sentence length information from $P_{M_{ID}}(\mathbf{x})$. To validate this hypothesis, we directly remove the sentence length information from $\log P_{M_{ID}}(\mathbf{x})$ by dividing it with the sentence length $|S|$, which forms $\log LN$. As shown in Table IV, LN shows very small correlation with the sentence length and its histogram in Figure 1D shows even less overlap between ID and OOD samples compared with $LR_{ws}$. This comparison explains the two advantages of LN over $LR_{ws}$, i.e., its comparable or even superior OOD detection performance and less computation and complexity.

### B. Ablation Study

We did an ablation study by removing either $\mathcal{L}_{cluster}$ or $\mathcal{L}_{CL}$ when optimizing Equation 4 for the representation learning based methods. These results are included in Table III. By comparing with the full solution, we see that: (1)

TABLE IV
PEARSON CORRELATION COEFFICIENT AND P-VALUE BETWEEN SENTENCE LENGTH AND LIKELIHOOD FOR ALL FOUR DATASETS.

| | ROSTD | | SNIPS | | Stackoverflow | | Searchsnippets | |
|---|---|---|---|---|---|---|---|---|
| | r | p-value | r | p-value | r | p-value | r | p-value |
| $P_{ID}(x)$ | -0.566 | 0.000 | -0.732 | $3.279 \times 10^{-118}$ | -0.803 | 0.000 | -0.827 | 0.000 |
| $P_B(x)$ | **-0.785** | 0.000 | **-0.866** | $1.378 \times 10^{-212}$ | **-0.945** | 0.000 | **-0.950** | 0.000 |
| $LR_{ws}$ | 0.014 | 0.122 | -0.225 | $1.762 \times 10^{-9}$ | -0.570 | $1.913 \times 10^{-172}$ | -0.176 | $3.080 \times 10^{-39}$ |
| LN | -0.026 | 0.005 | -0.099 | 0.008 | 0.044 | 0.049 | -0.053 | $8.461 \times 10^{-5}$ |

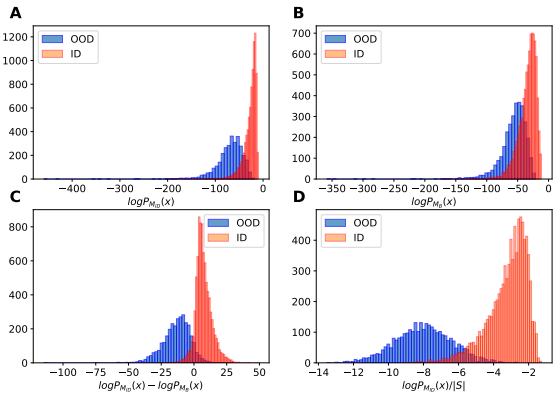


Fig. 1. Histogram of four kinds of likelihood for the ROSTD dataset with LSTM as the base model. A: $\log P_{M_{ID}}(x)$; B: $\log P_{M_B}(x)$; C: $\log LR_{ws}$; D: $\log LN$.

Both training objectives contribute to the improvement. (2) In some cases (e.g., Stackoverflow and Searchsnippets datasets), removing $\mathcal{L}_{CL}$ would cause larger performance degradation, while in the other cases, $\mathcal{L}_{cluster}$ is more important. This is dependent on the distributions of the ID samples (separation of different classes) and how OOD data is separated from ID data. We will show visualizations analysis of data later.

### C. Influence of $K$

Since we do not have any information on the ID labels, we need to tune the number of clusters $K$ by optimizing the OOD detection performance metric $AUPR_{OOD}$ on the validation set. Table V shows these values for different data sets along with their numbers of classes for the ID data for the representation learning based methods. We can see that the optimal number of clusters for OOD detection is not equal to the actual number of labels within ID data. For example, the optimal $K$ is 30 for the Searchsnippets dataset, which is far away from the number of labels, i.e., 6. This implies that our method is not simply an unsupervised approach to learning the classification task but rather a method more suitable for learning ID data distribution.

### D. Visualization of Representations

To obtain a qualitative sense of how our proposed representation learning method works, we provide T-SNE visualization plots of sentence representations for both ID and OOD samples

TABLE V
COMPARISON BETWEEN THE OPTIMAL NUMBER OF CLUSTERS AND THE NUMBER OF ID LABELS. SINCE THREE DATASETS HAVE 5 RANDOM SPLITS, WE LIST BOTH MINIMUM AND MAXIMUM NUMBER OF LABELS WITHIN THE PARENTHESES.

| | ROSTD | SNIPS | Stackoverflow | Searchsnippets |
|---|---|---|---|---|
| Optimal $K$ | 14 | 6 | 17 | 30 |
| Num of labels | 12 | (5, 5) | (13, 14) | (6, 6) |

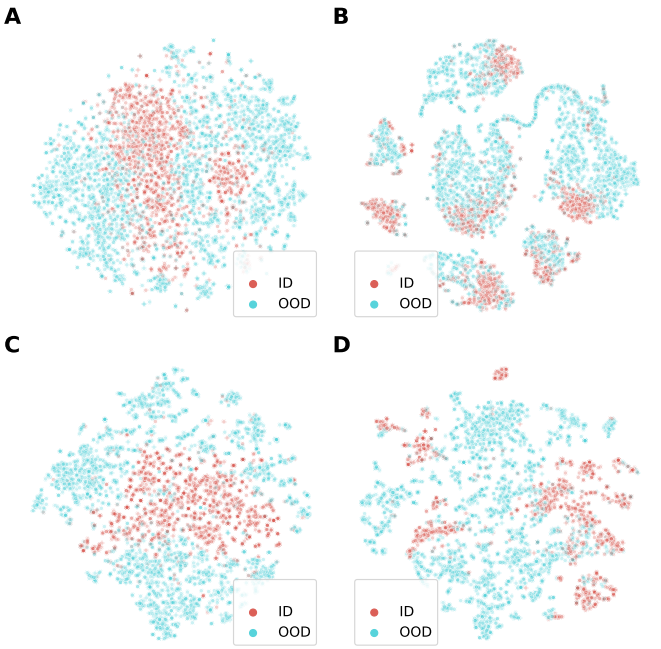


Fig. 2. T-SNE visualization of learned sentence representations for both ID and OOD samples of Searchsnippets dataset. A: DistilBERT-Base-NLI; B: DistilBERT-Base-NLI fine-tuned via unsupervised clustering; C: DistilBERT-Base-NLI fine-tuned via contrastive learning; D: DistilBERT-Base-NLI fine-tuned via combining clustering and contrastive learning.

of the Searchsnippets, SNIPS, Stackoverflow, and ROSTD datasets in Figure 2, 3, 4, and 5, respectively. As can be seen, the unsupervised clustering module can aggregate data points into many clusters by bringing close sentences of the same classes while pushing away sentences of different classes (e.g., Figure 2B, 3B, 4B, and 5B). In contrast, the contrastive learning module can distribute data points uniformly over the whole latent space, in which process ID and OOD samples can be pushed away from each other (e.g., Figure 2C). By combining these two modules, we can realize both goals and make the boundaries between ID and OOD samples much

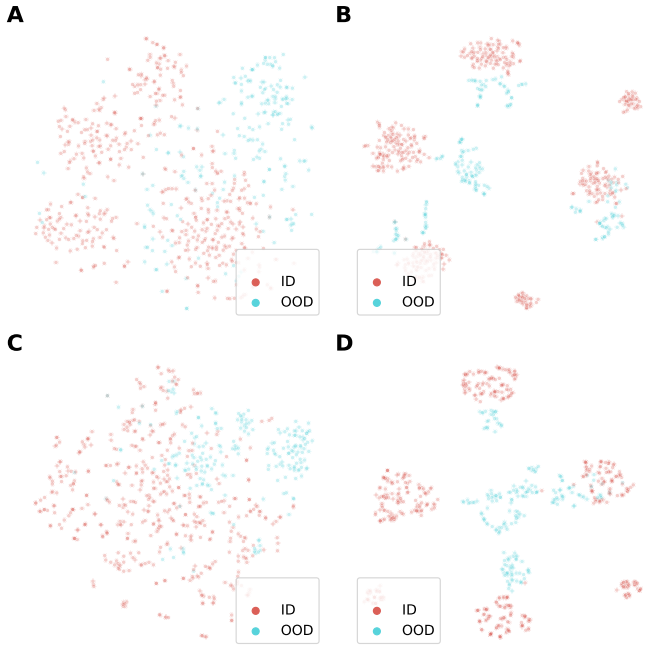


Fig. 3. T-SNE visualization of learned sentence representations for both ID and OOD samples of SNIPS dataset. A: DistilBERT-Base-NLI; B: DistilBERT-Base-NLI fine-tuned via unsupervised clustering; C: DistilBERT-Base-NLI fine-tuned via contrastive learning; D: DistilBERT-Base-NLI fine-tuned via combining clustering and contrastive learning.

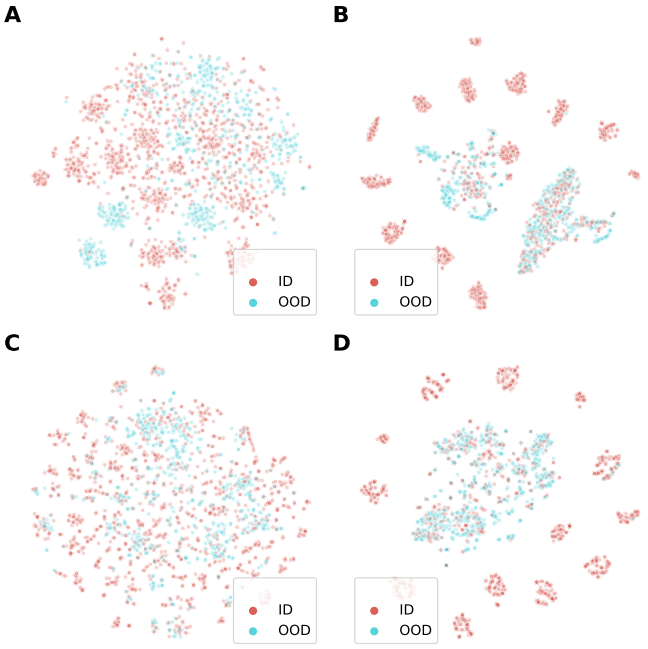


Fig. 4. T-SNE visualization of learned sentence representations for both ID and OOD samples of Stackoverflow dataset. A: DistilBERT-Base-NLI; B: DistilBERT-Base-NLI fine-tuned via unsupervised clustering; C: DistilBERT-Base-NLI fine-tuned via contrastive learning; D: DistilBERT-Base-NLI fine-tuned via combining clustering and contrastive learning.

more clear compared with no adaptive fine-tuning (e.g., Figure 2A vs. Figure 2D), and thus achieve the optimal OOD detection performance. Note when both modules are combined, the OOD samples tend to be condensed into the center, while ID samples are scattered outside of OOD samples in clusters (e.g., Figure 2D, 3D, 4D, and 5D), which facilitates the density estimator to better differentiate ID and OOD samples.

More specifically, for the Searchsnippets dataset, without the help of contrastive learning, representations obtained via only clustering still cannot well differentiate OOD samples from ID samples because ID and OOD samples still mingle with each other for some clusters (see Figure 2B). In contrast, from Figure 2C, we find that contrastive learning alone can better separate ID and OOD samples than no adaptation by condensing the ID data distribution (red dots) and spreading OOD samples outside (blue dots). In this case, the contrastive learning module contributes more to the full model performance than the clustering module due to its better capability at differentiating ID and OOD samples, which aligns with the conclusion drawn from the ablation study in Table III. However, for the ROSTD and SNIPS datasets, unsupervised clustering itself can already well separate ID and OOD samples by distributing the ID samples into many condensed clusters, where OOD samples are distributed outside of the ID clusters. This explains the finding revealed by the Ablation study in Section VI-B that the clustering objective contributes more to the full model performance.

## VII. Conclusion

In this work, we aim at the OOD detection problem without using any ID labels. In addition to evaluating different LM based likelihood methods, we propose a representation learning based method by performing unsupervised clustering and contrastive learning to learn good data representations for OOD detection. We demonstrate that this novel unsupervised method can not only outperform the best likelihood based methods but also be even competitive to the state-of-the-art supervised method that has extensively used labeled data.

## References

[1] H. Xu, K. He, Y. Yan, S. Liu, Z. Liu, and W. Xu, "A deep generative distance-based classifier for out-of-domain detection with mahalanobis space," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 1452–1460.

[2] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[3] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations*, 2017.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[5] L. Shu, H. Xu, and B. Liu, "DOC: Deep open classification of text documents," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2911–2916. [Online]. Available: https://www.aclweb.org/anthology/D17-1314

[6] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ryiAv2xAZ

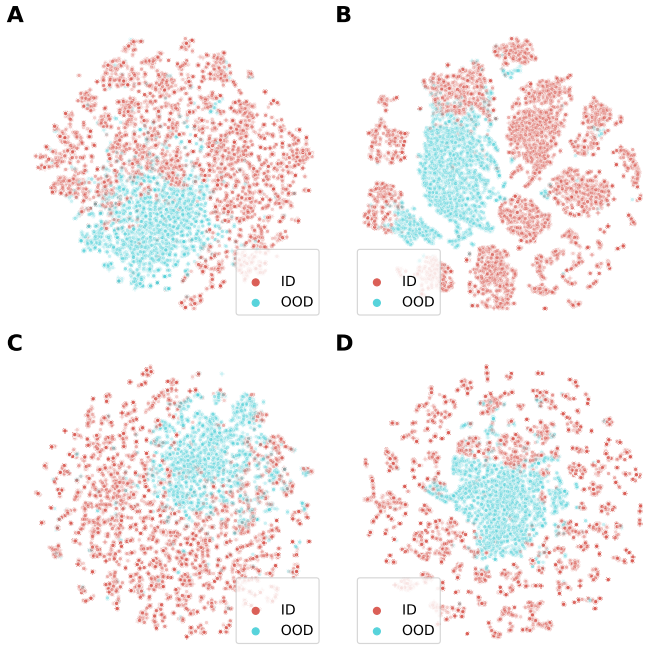


Fig. 5. T-SNE visualization of learned sentence representations for both ID and OOD samples of ROSTD dataset. A: DistilBERT-Base-NLI; B: DistilBERT-Base-NLI fine-tuned via unsupervised clustering; C: DistilBERT-Base-NLI fine-tuned via contrastive learning; D: DistilBERT-Base-NLI fine-tuned via combining clustering and contrastive learning.


[7] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[8] A. Podolskiy, D. Lipin, A. Bout, E. Artemova, and I. Piontkovskaya, "Revisiting mahalanobis distance for transformer-based out-of-domain detection," *arXiv preprint arXiv:2101.03778*, 2021.

[9] J. Zhang, K. Hashimoto, W. Liu, C.-S. Wu, Y. Wan, P. Yu, R. Socher, and C. Xiong, "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)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5064–5082. [Online]. Available: https://aclanthology.org/2020.emnlp-main.411

[10] C. Xia, C. Zhang, X. Yan, Y. Chang, and P. Yu, "Zero-shot user intent detection via capsule neural networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3090–3099. [Online]. Available: https://aclanthology.org/D18-1348

[11] M. Namazifar, A. Papangelis, G. Tur, and D. Hakkani-Tür, "Language model is all you need: Natural language understanding as question answering," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7803–7807.

[12] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[13] V. Gangal, A. Arora, A. Einolghozati, and S. Gupta, "Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7764–7771.

[14] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, and M. Kloft, "Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4061–4071. [Online]. Available: https://www.aclweb.org/anthology/P19-1398

[15] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, "Learning and evaluating representations for deep one-class classification," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=HCSgyPUfeDj

[16] H. Choi and E. Jang, "Generative ensembles for robust anomaly detection," 2019. [Online]. Available: https://openreview.net/forum?id=B1e8CsRctX

[17] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=H1xwNhCcYm

[18] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HyxCxhRcY7

[19] M. Tan, Y. Yu, H. Wang, D. Wang, S. Potdar, S. Chang, and M. Yu, "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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3566–3572. [Online]. Available: https://www.aclweb.org/anthology/D19-1364

[20] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=H1VGkIxRZ

[21] Y. Zheng, G. Chen, and M. Huang, "Out-of-domain detection for natural language understanding in dialog systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1198–1209, 2020.

[22] R. Kamoi and K. Kobayashi, "Why is the mahalanobis distance effective for anomaly detection?" *arXiv preprint arXiv:2003.00402*, 2020.

[23] H. Xu, K. He, Y. Yan, S. Liu, Z. Liu, and W. Xu, "A deep generative distance-based classifier for out-of-domain detection with mahalanobis space," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1452–1460. [Online]. Available: https://www.aclweb.org/anthology/2020.coling-main.125

[24] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[26] D. Zhang, F. Nan, X. Wei, S. Li, H. Zhu, K. McKeown, R. Nallapati, A. Arnold, and B. Xiang, "Supporting clustering with contrastive learning," *arXiv preprint arXiv:2103.12953*, 2021.

[27] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[28] D. Jurafsky, *Speech & language processing*. Pearson Education India, 2000.

[29] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[30] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016, pp. 478–487. [Online]. Available: http://proceedings.mlr.press/v48/xieb16.html

[31] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[32] T.-E. Lin and H. Xu, "Deep unknown intent detection with margin loss," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5491–5496. [Online]. Available: https://www.aclweb.org/anthology/P19-1548

[33] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3795–3805. [Online]. Available: https://www.aclweb.org/anthology/N19-1380

[34] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Networks*, vol. 88, pp. 22–31, 2017.

[35] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 91–100.

[36] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[37] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[39] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[40] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.