THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

TRANSFORMER-BASED TEXTUAL OUT-OF-DISTRIBUTION DETECTION: METHODS AND ANALYSIS

ZHAN LIMING

PhD

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University

Department of Computing

Transformer-Based Textual Out-of-Distribution Detection: Methods and Analysis

ZHAN LIMING

A thesis submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

March 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

ZHAN Liming (Name of student)

Abstract

The success of machine learning methods heavily relies on the assumption that the test data follows a similar distribution to the training data. However, this assumption is frequently violated in real-world scenarios. Detecting distribution shifts between training and inference, referred to as out-of-distribution (OOD) detection, is crucial to prevent models from making unreliable predictions. OOD detection is particularly significant in ensuring the safe use of deep neural networks. Despite its importance and the surge of research in the vision domain, this problem is often overlooked in natural language processing (NLP).

This thesis aims to address this gap by proposing and evaluating novel transformer-based OOD detection approaches for various NLP classification tasks, such as dialogue intent detection, topic classification, sentiment classification, and question classification.

First, we present an efficient end-to-end learning framework to reduce the complexity of training textual OOD detectors. Since the distribution of OOD samples is arbitrary and unknown in the training stage, previous methods commonly rely on strong assumptions on data distribution such as mixture of Gaussians to make inference, resulting in either complex multi-step training procedures or hand-crafted rules such as confidence threshold selection for OOD detection. To develop a simplified learning paradigm for textual OOD detection, we propose to train a (K+1)-way discriminative classifier by simulating the test scenario during training. Specifically, we construct a set of pseudo OOD samples in the training stage, by generating synthetic OOD samples using in-distribution (ID) features via self-supervision and sampling OOD sentences from easily

available open-domain datasets. The pseudo outliers are used to train a discriminative classifier that can be directly applied to and generalize well on the test task.

Second, we address the challenge of low-resource settings for textual OOD detection, a critical problem often encountered in the development of machine learning systems. Despite its significance, this problem has received limited attention in the literature and remains largely unexplored. We conduct a thorough investigation of this problem and identify key research issues. Through our pilot study, we uncover why existing textual OOD detection methods fall short in addressing this issue. Building on these findings, we propose a promising solution that leverages latent representation generation and self-supervision.

Finally, we delve into Transformer-based representation learning for textual OOD detection. Existing methods commonly adopt the discriminative training objective – maximizing the conditional likelihood $p(y|x)$ – which is biased and leads to suboptimal OOD detection performance. To address this issue, we propose a generative training framework based on variational inference, which directly optimizes the likelihood of the joint distribution $p(x, y)$. Specifically, our framework takes into account the unique characteristics of textual data and leverages the representations of pre-trained Transformers in an efficient manner.

In summary, this thesis provides novel and effective Transformer-based approaches to address the challenges of textual OOD detection. Our proposed methods show significant improvements over existing state-of-the-art methods, and our findings can have practical applications in improving the robustness of machine learning models in NLP.

Publications arising from the thesis

1. Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, Albert Y.S. Lam. "Out-of-Scope Intent Detection with Self-Supervision and Discriminative Training", In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL-IJCNLP 2021, Association for Computational Linguistics. (Oral Presentation)

2. Li-Ming Zhan, Haowen Liang, Lu Fan, Xiao-Ming Wu, Albert Y.S. Lam. "A Closer Look at Few-Shot Out-of-Distribution Intent Detection", In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2022. (Oral Presentation)

3. Li-Ming Zhan, Bo Liu, Zexin Lu, Xiao-Ming Wu. "Transformer-Based Textual Out-of-Distribution Detection with Variational Inference", under review.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Wu Xiao-Ming, for her guidance, support, and encouragement throughout my PhD study. Her profound knowledge, insightful comments, and constructive criticism have greatly enriched my research and helped me grow as a researcher. Her unwavering dedication to academic excellence and her passion for advancing scientific knowledge have been an endless source of inspiration for me. I feel extremely fortunate to have had the opportunity to work with such an outstanding mentor and researcher.

I would like to extend my sincere gratitude to all my lab mates, friends, and collaborators who have contributed to my research and provided me with valuable feedback and support throughout my PhD journey. In particular, I would like to thank Jiaxin Chen, Bo Liu, Lu Fan, Haowen Liang, Qimai Li, Li Xu, Zexin Lu, Han Liu, and Sihan Wang for their friendship, encouragement, and intellectual stimulation. Your insights, discussions, and critiques have made my PhD experience a memorable and rewarding one.

Finally, I would like to extend my sincere thanks to my family for their love, encouragement, and unwavering support throughout my academic journey. Their constant belief in me, and their sacrifices and understanding during difficult times have been invaluable. Without them, I would not have been able to pursue my dreams and reach this milestone.

Thank you all for making this journey possible.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decade, there has been significant progress in the development of machine learning (ML) models, particularly deep neural networks (DNNs), thanks to advances in infrastructure, such as the accelerated computation speed of GPUs, and architecture, such as the large-scale Transformers. Consequently, DNN-based applications have become increasingly prevalent in our daily lives, including machine conversation systems (e.g., Microsoft's Cortana and Apple's Siri) and image recognition systems (e.g., systems based on facial recognition and pedestrian detection). However, ensuring the reliability of these models is a key concern in ML safety (Hendrycks et al., 2021b), especially for safety-critical applications such as medical diagnosis and autonomous driving systems. Reliability can be interpreted as the confidence of the model in its predictions and the likelihood of failure.

Unfortunately, it has been found that DNNs are often too confident in their predictions when presented with anomalous or out-of-distribution (OOD) inputs, which limits their applicability in high-stakes settings. Their vulnerability to OOD inputs is largely due to distribution shifts between the training and test distributions. To make a ML model feasible for training and development, it is typically necessary to assume that the training and test distributions are independent and identically distributed (i.i.d.). However, this closed assumption is often not valid in realistic scenarios, where the inherent complexity of real-world data makes it infeasible to collect enough training data to fully capture the target distribution and account for unknown unknowns.

In response to this challenge, OOD detection with DNNs has garnered substantial attention in the past six years. This technique aims to enhance a model's resilience when faced with semantically unknown inputs that exhibit significant distribution shifts. Specifically, when applied to a given target task, OOD detection considers the training data for that task to be in-distribution (ID), while realistic OOD data is unbounded and not available during training. The primary objective is to accurately differentiate between ID and OOD inputs while simultaneously addressing the ID target task.

OOD detection is intensively discussed in the context of computer

vision tasks with the flourish of convolutional neural networks (CNNs). However, despite its crucial importance in numerous natural language processing (NLP) applications, such as identifying spam and fake news, detecting OOD intents in dialogue systems, and preventing ethical dilemmas, textual OOD detection has not yet received comparable attention. Meanwhile, the rise of Transformer-based pre-trained language models (PLMs) has driven significant advancements in a range of mainstream NLP tasks over the past several years. Nonetheless, current research on textual OOD detection are still limited to the direct application of general methods or transfer methods from the visual domain to NLP tasks, without fully accounting for the intrinsic properties of textual data and the semantic power of Transformers. The research in this thesis endeavors to bridge this gap by conducting a systematic and rigorous study of textual OOD detection with PLMs.

1.1 Challenges

Challenge I: Impractical implementation procedure. Popular methods tackle the OOD detection problem by developing a specific confidence score based on a model trained on in-distribution (ID) data. Subsequently, a manually chosen threshold on the confidence score is employed to differentiate between ID and OOD inputs. Although this

post-threshold strategy allows for flexibility in controlling the sensitivity and specificity of the binary OOD detector, it requires expert knowledge and additional validation data to select a suitable threshold. Moreover, since the deployment environment is subject to change, manual updates to the threshold could be necessary, making the process case-specific. Abandoning the threshold process would require a departure from the existing learning paradigm for OOD detection, which is challenging given the unrepresentable nature of actual OOD training data.

Challenge II: Low resource. The process of designing and collecting large-scale labeled data is a demanding and resource-intensive task. The difficulty is further amplified in textual tasks due to the subjective nature of semantics, making consistently defining and labeling textual data more difficult than vision data. Consequently, in real-world scenarios, textual OOD detection often faces low resource challenges, where the labeled ID dataset contains only a few examples per class. As the parameters of DNNs increase, training or fine-tuning them on a small labeled dataset could result in significant overfitting, which is even more pronounced in textual OOD detection given the large number of parameters in current pre-trained Transformers (often billions). For instance, state-of-the-art OOD intent detectors perform worse than random guessing in low-resource scenarios. Therefore, it is worth exploring

low-resource textual OOD detection in more depth.

Challenge III: Biased training objective. Commonly, OOD detection models update their parameters by minimizing the ID discriminative loss with respect to the conditional probability $p(y \mid x)$, since real OOD data is challenging to represent. The binary ID *vs.* OOD classification is performed based on a heuristic statistic derived from the model, which can serve as an empirical indicator of the model's confidence. However, the ID discriminative loss is tailored to optimize the model only for the ID task, without explicitly accounting for the detection of anomalous unknowns. Although some empirical confidence scores have shown effectiveness, re-designing the training stage of OOD detection methods to directly target the task goal can be a more promising and fundamental approach than relying solely on heuristic methods. To the best of our knowledge, no prior research has systematically investigated this direction.

1.2 Contributions

In this research, we aims to address these aforementioned challenges of textual OOD detection. To this end, we undertake a thorough investigation of the potential of contextualized representations from pre-trained Transformers to reform the learning paradigm and streamline the de-

ployment process of textual OOD detection. Our study gives rise to innovative learning paradigms that significantly improve the efficiency and effectiveness of textual OOD detection. Moreover, our proposed methods primarily align with the principles of self-supervised and generative learning, thereby circumventing the requirement for additional training data and exhibiting remarkable generalization capabilities. Our proposed methods have been applied to a wide range of NLP tasks, including OOD intent detection in dialogue systems, topic classification, sentiment analysis, and question classification. The contributions of this thesis are summarized as follows:

Contribution 1: Self-supervised OOD Detection. To tackle Challenge I, we present a novel end-to-end learning framework that simplifies traditional OOD detection approaches by eliminating the manual threshold selection requirement. Our method can yield a model that is readily applicable to OOD detection tasks after training. Transformers have been proved to be extremely effective in dealing with various NLP tasks due to the high contextualization level of the resulting textual representation. Our work leverages the representation space of Transformers to generate OOD representations by convex combinations between ID representations. These OOD representations enable the training of a

(K+1)-way discriminator, which models OOD data as an abstract class and can classify an input as belonging to either one of the K ID classes or the OOD class. We verify the effectiveness and explore the properties of our framework in the context of OOD intent detection, a commonly studied textual OOD detection task. Our research has been accepted in ACL2021 (Zhan et al., 2021).

Contribution 2: Low-resource OOD Detection. For Challenge II, we investigate textual OOD detection in low-resource settings and highlight the insufficiency of current techniques in such scenarios. To this end, we propose a novel approach that involves learning a latent denoising autoencoder (DAE) in the representation space of Transformers. In our model, a lightweight DAE is learned in a self-supervised manner and can enhance the ID dataset by generating samples around the vicinity of ID representations. As a classic generative model, DAE offers sampling efficiency and guarantees consistency in approximating the target distribution. We find that the DAE can be easily trained to capture the distribution of ID representations. In addition, the proposed method is compatible with our aforementioned (K+1)-way training framework. We demonstrate our proposed approach could significantly improve the performance of low-resource OOD intent detection tasks. The proposed

work has been published in COLING2022 (Zhan et al., 2022).

Contribution 3: A Unified Probabilistic Framework. To address Challenge III, we propose a principled learning framework to learn better representations for textual OOD detection. Most existing OOD detection methods directly operate on the output of the model's last layer. However, the upper layers of Transformers are more geared towards producing ID task-specific representations (Ethayarajh, 2019) that are sub-optimal for OOD discrimination. To address this, we propose optimizing the model with respect to the joint distribution $p(x, y)$ instead of $p(y|x)$. By doing so, we aim to preserve the information in x that can benefit OOD detection. We believe that this information can serve as useful evidence for OOD discrimination, even though it is not relevant to ID discrimination. We use an amortized variational Bayesian inference (VI) learning strategy to make the objective $p(x, y)$ tractable. We also redesign the original VI architecture to better leverage the intermediate representations in Transformers. We demonstrate the effectiveness of our proposed framework on various NLP tasks and show that it can significantly improve the performance of state-of-the-art OOD detectors by learning better representations.

Thesis organization. In Chapter 2, we provide an overview of the background of this thesis, covering topics such as the importance of OOD detection, general OOD detection methods, textual OOD detection methods, and the development of Transformer-based models. In Chapter 3, we present our self-supervised learning framework for addressing Challenge I. Chapter 4 delves into Challenge II and proposes a lightweight ID data augmentation method based on latent denoising autoencoders. For Challenge III, discussed in Chapter 5, we invent a novel probabilistic representation learning framework for textually OOD detection. Finally, in Chapter 6, we conclude this thesis and explore several potential directions for future research.

Chapter 2

Background and Related Work

This chapter begins with an introduction to out-of-distribution (OOD) detection, along with its associated research areas, such as OOD generalization, anomaly detection, and more. Additionally, we provide a brief overview of the evolution of Transformer-based models.

2.1 Foundations

Machine learning safety. The goal of machine learning (ML) safety is to develop measures and algorithms that can steer ML systems in a reliable and safe direction, enabling them to withstand the complexity of real-world environments. In their efforts to provide clarity and direction to the research community, Hendrycks et al. (2021b) have characterized

ML safety into two categories: *reliability* and *alignment*. Reliability investigates the out-distribution properties of ML models, while alignment is concerned with aligning ML models with human values, including ethical considerations. Although reliability has garnered significant attention from the research community, alignment is still in its infancy due to the difficulty of defining and specifying human values in a way that machines can understand. With recent advancements in artificial general intelligence, such as diffusion models (Saharia et al., 2022; Rombach et al., 2022) and ChatGPT (Ouyang et al., 2022), alignment is expected to become an increasingly important focus. In this study, however, our focus is on the OOD detection task within the reliability branch.

Distribution shift. Reliability in the realm of machine learning focuses on endowing models with the capability to handle unforeseen circumstances that were not represented in the training data. It can be segregated into two specific objectives, namely OOD generalization and OOD detection. The key difference between these objectives lies in the extent of distribution shift. The shift can occur in the marginal distribution $p(x)$ or in both $p(y)$ and $p(x)$. If the shift occurs only in the input space $\mathcal{X}$, it is referred to as *covariate shift*; otherwise, it is called *semantic shift* (Yang et al., 2021).

OOD generalization. The objective of OOD generalization, also known as OOD robustness, is to improve the performance of ML models in the presence of covariate shift. Covariate shift is a well-defined concept in the field of computer vision and includes phenomena such as adversarial examples (Goodfellow et al., 2015), domain shift (Quinonero-Candela et al., 2008), changes in image style, blurriness, geographic location, camera operation, and more. Recently, there has been significant progress in visual OOD generalization, owing to the availability of more specific and diverse benchmarks. For instance, Hendrycks et al. (2021a) introduced four datasets for visual OOD generalization, which consider visual renditions (ImageNet-R), changes in the image capture process (StreetView StoreFronts and DeepFashion Remixed), and natural blurry effects (ImageNet-C). However, the subjective nature of textual data hinders the development of textual OOD generalization research to the same level of granularity as the visual domain. Textual OOD generalization is still limitedly studied in the domain shift scenarios. For instance, Hendrycks et al. (2020) suggest evaluating the performance of textual OOD generalization using sentiment analysis datasets from various domains, such as Amazon reviews from different product categories.

OOD detection. The objective of OOD detection is to address the issue of semantic shift. In scenarios where the model is exposed to samples from unfamiliar semantic classes, OOD detection mandates the model to identify these inputs as OOD for further human inspection or to apply a reliable fallback approach. Conversely, when the model is faced with samples from the in-distribution (ID) classes that were encountered during training, it must also predict their corresponding classes with high accuracy. OOD detection is the focus of this thesis and will be elaborated in the next section.

Related topics. In the final part of this section, we discuss the distinctions between OOD detection and other related subjects. **Anomaly detection** (Ruff et al., 2021) targets identifying anomalous samples that deviate from the predetermined normality, which may stem from either covariate shift or semantic shift. The main distinction between anomaly detection and OOD detection is that OOD detection necessitates multi-class ID discrimination, while anomaly detection does not. **Novelty detection** (Markou and Singh, 2003) is comparable to anomaly detection but considers ”abnormal” samples as ”novel” discoveries. On the other hand, **outlier detection** (Aggarwal and Yu, 2001) belongs to transductive learning, whereas OOD detection is inductive. Outlier detection is

intended to detect an outlier based on the entire set of observations.

2.2 Out-of-distribution Detection

This section provides a detailed discussion of commonly used approaches and benchmarks for out-of-distribution (OOD) detection. While many of these methods were originally developed for OOD detection in computer vision, some of these methods have also been adapted for textual OOD detection. In this section, we will also discuss these representative methods and their applications in textual OOD detection.

2.2.1 Post-hoc Methods

Post-hoc approaches are the most widely used methods for OOD detection. These methods involve deriving a statistic from a trained in-distribution (ID) model that can serve as a measure of predictive confidence. The key advantages of post-hoc methods are that they are model-agnostic and do not require additional gradient updates, which have contributed to their widespread adoption for OOD detection.

The pioneering work MSP (Hendrycks and Gimpel, 2017) proposes to use the maximum softmax probability as the ID confidence score. It is the first work for OOD detection with deep neural networks (DNNs) and defines the evaluation protocols for OOD detection. Hendrycks and

Gimpel (2017) also apply MSP to textual OOD detection task. MSP has been the most famous baseline for OOD detection. Thereafter, OOD detection has attracted increasing attention in the community. The following work ODIN (Liang et al., 2018) proposes to use temperature scaling and add small perturbations to inputs to obtain more separable sfotmax outputs for in- and out-of-distribution images. Lee et al. (2018b) propose to use the input's minimal Mahalanobis distance (MD) with respect to ID class centroids. MD is widely applied for visual and textual OOD detection and among the best methods for a long time (Ren et al., 2021). Liu et al. (2020) propose to employ energy-based OOD scores derived from the logits of the softmax layer to discriminate ID and OOD examples.

Besides these OOD scores derived around the softmax layer (top and penultimate layers of DNNs), following works take deeper steps to investigate the intermediate layers and gradients of DNNs. For example, GRAM (Sastry and Oore, 2020) proposes to compute Gram matrices of hidden states and identify OOD examples by comparing the range of Gram matrix values with the respective observations over the training data. GradNorm (Huang et al., 2021) finds out that the vector norm of parameter gradients (backpropagated from the KL divergence between the softmax output and a uniform probability) could be an effective

indicator for predictive confidence. ReAct (Sun et al., 2021) points out that the intermediate activations of DNNs exhibit differently between ID and OOD examples and the rectified activations make ID and OOD more separable. distribution.

2.2.2 Uncertainty Estimation Methods

In addition to these post-hoc OOD detection methods discussed above, the second line of OOD detection methods takes an unsupervised perspective and tries to address the OOD detection problem by solving a more general problem – density estimation. Their intuition is that learning as much as possible knowledge about the density of the training ID distribution can help us to solve any problem related to the data (Schölkopf et al., 1999). Their learning target is the density function of the training set – $p_{ID}(x)$ – such that OOD examples are assumed to yield lower probabilities than the ID ones. However, in high dimensional spaces, this assumption does not hold in practice and many previous works (Choi et al., 2018) have found that OOD examples may be assigned higher likelihoods than ID examples. Recent works (Ren et al., 2019; Nalisnick et al., 2019; Morningstar et al., 2021) are still trying to correct this pathology. It is worth mentioning that according to Morningstar et al. (2021), the likelihoods in high-dimensional spaces could be affected by

other state configurations besides the probability, and they propose to address the issue by conditioning OOD detection on multiple summary statistics.

2.2.3 Data Augmentation Methods

Data augmentation methods in the vision domain have made significant progress in recent times and have demonstrated a more promising approach for OOD detection than the classical solutions mentioned earlier. For example, the Mixup approach, as observed by (Thulasidasan et al., 2019), may result in models that could produce better softmax probabilities that are more consistent with the true likelihood of an accurate prediction. The Outlier Exposure (OE) method proposed by Hendrycks et al. (2019) suggests using carefully chosen representative OOD data to expose the model to OOD learning signals. Similarly, Meinke and Hein (2020) adopt the same approach of utilizing additional OOD data. Specifically, OE minimizes the KL divergence between the softmax probability of OOD data and a uniform distribution. PixMix (Hendrycks et al., 2022b) is the new sate-of-the-art visual OOD detection method though comprehensive evaluations(Yang et al., 2022). PixMix has been proposed as a solution not only for OOD detection but also for other AI safety tasks such as OOD generalization and prediction consistency. It is

the first approach that has demonstrated competitive performance across all AI safety tasks. Despite its simplicity, PixMix is an effective data augmentation technique that blends an original image with complex fractal images. The authors point out that the inherent structural complexity of fractals can substantially improve model reliability.

2.2.4 Benchmarks and Metrics

Threshold-independent evaluation metrics include AUROC, FPR@95 and AUPR are frequently used in OOD detection (Yang et al., 2022). Recently, Khosla and Gangadharaiah (2022) argue that threshold-dependent metrics, such as detection accuracy and F1-scores, may offer more insight into the OOD detector's generalization ability. The following chapters will provide more details on these metrics.

While benchmarks for visual OOD detection have flourished, there is currently a lack of datasets specifically designed for textual OOD detection. MNIST (Mu and Gilmer, 2019), ImageNet (Krizhevsky et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton, 2009) are commonly used benchmark datasets for visual OOD detection. In order to make visual OOD detection applicable in large-scale real-world scenarios, a recent proposal by Hendrycks et al. (2022a) introduces the *Species dataset* that comprises thousands of classes and complex scenes.

This dataset serves as a strong basis for the further development of visual OOD detection for practical purposes.

There has been less progress on developing benchmarks for textual OOD detection compared to the visual domain. CLINC150 (Larson et al., 2019) is an OOD intent detection dataset for the task of detecting OOD utterances in dialogue systems. Hendrycks et al. (2020) evaluate the textual OOD detection performance of Transformers by taking SST-2 (Socher et al., 2013) as the ID dataset and five additional datasets from various NLP tasks as OOD datasets. Zhou et al. (2021) further extend it by including more NLP tasks such as topic classification, question classification and natural language inference. However, creating benchmark datasets that are specifically designed for evaluating textual OOD detection would be a significant step forward, similar to the impact that the Species dataset had in the vision domain.

2.3 Textual OOD Detection

The significance of textual OOD detection in ensuring the robustness of NLP applications, such as dialogue systems, has led to a surge in research interest. A classical OOD detection task in the textual domain is OOD intent detection in dialogue systems, which requires detecting utterances with unknown intents. In the general textual OOD detection context,

recent studies (**?**Podolskiy et al., 2021; Zhou et al., 2021) have explored the application of general or visual OOD methods to textual scenarios and investigated the OOD characteristics of Transformers. In Chapter 5, we present a general variational framework specifically designed for textual OOD detection. Throughout the remainder of this section, we will be introducing exemplary works that have addressed textual OOD detection.

The first line of works (Hendrycks et al., 2020; Shu et al., 2017; Ryu et al., 2018, 2017) uses some statistic as the confidence score of whether an example is OOD or not. Hendrycks and Gimpel (2017) point out that the negative probability outputted by the softmax function can be a good confidence metric for OOD detection. Shu et al. (2017) define a binary classification task for every in-domain class and used the maximum probability among all these binary classifiers as the confidence score. Ryu et al. (2018) develop an adversarial training strategy inspired by GAN for OOD intent detection. The discriminator in GAN was trained to assign lower scores to OOD examples. Ryu et al. (2017) employ an autoencoder trained on in-domain examples and used the reconstruction score as the OOD indicator. However, all these methods require manual effort in selecting a proper threshold for OOD discrimination.

The second line of works (Lin and Xu, 2019; Zhang et al., 2021; Yan

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

Different from previous methods, our work presented in Chapter 3 propose to directly learn a $(K+1)$-way classifier in an end-to-end manner. We create OOD learning signals during training by leveraging external data or constructing simulated OOD examples with self-supervised information.

Moreover, few-shot textual OOD detection is under-explored and has never been investigated in a strictly low-resourced setting. The most related work is DNNC proposed in Zhang et al. (2020), which tries to mitigate the data-scarcity problem in the OOD intent detection task by fine-tuning RoBERTa on external large natural language inference datasets. In Chapter 4, however, we consider using the few-shot labeled examples as the only training resource.

2.4 Transformer-based Pre-trained Language Models

Pre-trained language models have become a popular and effective approach for natural language processing (NLP) tasks in recent years (Raffel et al., 2020). These models are typically trained on massive amounts of text data to general contextualized text representations, which can then be fine-tuned on specific downstream tasks such as sentiment analysis, named entity recognition and natural language inference.

Learning contextualized text representation has been a fundamental target in NLP. Early works(Mikolov et al., 2013; Pennington et al., 2014) have made it possible to represent words in a continuous space. However, these methods result in *static* word embeddings, without accounting for the specific context of the words. Thus a key problem arises from these methods is that all the senses of polysemous words are limited to share a fixed representation, which is not desirable in piratical. To this end, the pioneering work ELMo (Peters et al., 2018) proposes to use a pre-trained bidirectional LSTM to extract contextualized embeddings, allowing it to capture more nuanced meaning compared to traditional word embeddings. However, the architectures of ELMo are task-specific and need additional efforts to adapt cross various tasks.

Subsequently, Transformer-based pre-trained language models (Vaswani et al., 2017) start to be prominent in contextualized representation learning. These models can be classified into three categories based on their architecture.

The first line is encoder-based Transformers. The most representative work in this line is BERT (Devlin et al., 2019), namely **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. BERT proposes to pre-train a stack of Transformer encoder layers on a large set of corpus including BooksCorpus (Zhu et al., 2015) and English Wikipedia. Transformer-based architecture has strong capabilities in capturing the left and right context of word spans with the low-cost operation *attention*. Moreover, BERT proposes an unsupervised pre-training objective called "masked language model" (MLM). This objective involves randomly masking tokens in the input, and the goal is to predict the masked tokens at the output layer. BERT has been widely utilized as the representation extractor in almost all natural language understanding tasks, and it eliminates the requirement for task-specific engineering. The family of BERT models has flourished. For example, RoBERTa (Liu et al., 2019) propose to train the model longer with bigger batches and longer sequence to achieve better performance. ALBERT (Lan et al., 2020) reduce the parameter scale of BERT by factorized embedding parameterization and cross-layer

parameter sharing.

The second line is decoder-based Transformers that takes a language generation perspective. The most representative work is the GPT series (Brown et al., 2020; Radford et al., 2019, 2018). The GPT models train a multi-layer Transformer decoder using the autoregressive objective, and the resulting hidden states can serve as contextualized representations for downstream NLP tasks. In a recent development, InstructGPT (also known as GPT-3.5) has been utilized for open-domain conversation tasks, leading to the creation of ChatGPT, a chatbot that has been hailed as a significant step towards artificial general intelligence.

The third class of works pertains to encoder-decoder based Transformers, which employ the traditional Transformer architecture consisting of an encoder and a corresponding decoder. Models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2020) fall into this category. BART explores a flexible random corruption function to enhance masked language modeling, while T5 reformulates various NLP tasks such as machine translation, abstract summarization, and text classification as text-to-text tasks. Additionally, T5 proposes a large-scale corpus called C4. The findings in T5 have propelled the research of Transformer-based pre-trained language models to next level.

In this thesis, we focus on using BERT family models for our research

on textual OOD detection, as it is a natural language understanding task. However, it would be worthwhile to explore in-depth the models from the other two categories in future work.

Chapter 3

Self-supervised OOD Detection

3.1 Introduction

Conversational system is becoming an indispensable component in a variety of AI applications and acts as an interactive interface provided to users to improve user experience. Language understanding is essential for conversational systems to provide appropriate responses to users, and intent detection is usually the first step of language understanding. The primary goal is to identify diverse intentions behind user utterances, which is often formalized as a classification task. However, intent classes defined during training are inevitably inadequate to cover all possible user intents at the test stage due to the diversity and randomness of user utterances. Hence, out-of-scope (or unknown) intent detection is

essential, which aims to develop a model that can accurately identify known (seen in training) intent classes while detecting the out-of-scope classes that are not encountered during training.

Due to the practical importance of out-of-scope intent detection, recent efforts have attempted to solve this problem by developing effective intent classification models. In general, previous works approach this problem by learning *decision boundaries* for known intents and then using some confidence measure to distinguish known and unknown intents. For examples, LMCL (Lin and Xu, 2019) learns the decision boundaries with a margin-based optimization objective, and SEG (Yan et al., 2020) assumes the known intent classes follow the distribution of mixture of Gaussians. After learning the decision boundaries, an off-the-shell outlier detection algorithm such as LOF (Breunig et al., 2000) is commonly employed to derive confidence scores (Yan et al., 2020; Shu et al., 2017; Lin and Xu, 2019; Hendrycks and Gimpel, 2017). If the confidence score of a test sample is lower than a predefined threshold, it is identified as an outlier.

However, it may be problematic to learn decision boundaries solely based on the training examples of known intent classes. First, if there are sufficient training examples, the learned decision boundaries can be expected to generalize well on known intent classes, but not on the

unknown. Therefore, extra steps are required in previous methods, such as using an additional outlier detection algorithm at the test stage or adjusting the confidence threshold by cross-validation. On the other hand, if there are not sufficient training examples, the learned boundaries may not generalize well on both known and unknown intents. As a result, these methods often underperform when not enough training data is given. Hence, it is important to provide learning signals of unknown intents at the training stage to overcome these limitations.

In contrast to previous works, we adopt a different approach by explicitly modeling the distribution of unknown intents. Particularly, we construct a set of pseudo out-of-scope examples to aid the training process. We hypothesize that in the semantic feature space, real-world outliers can be well represented in two types: "*hard*" outliers that are geometrically close to the inliers and "*easy*" outliers that are distant from the inliners. For the "hard" ones, we construct them in a self-supervised manner by forming convex combination of the features of inliers from different classes. For the "easy" ones, the assumption is that they are very unrelated to the known intent classes, so they can be used to simulate the randomness and diversity of user utterances. They can be easily constructed using public datasets. For example, in our experiments, we randomly collect sentences from datasets of other NLP tasks such as

question answering and sentiment analysis as open-domain outliers.

In effect, by constructing pseudo outliers for the unknown class during training, we form a consistent $(K + 1)$ classification task (K known classes + 1 unknown class) for both training and test. Our model can be trained with a cross-entropy loss and directly applied to test data for intent classification and outlier detection without requiring any further steps. As shown in Figure 3.1 (better view in color and enlarged), our method can learn better utterance representations, which make each known intent class more compact and push the outliers away from the inliers. Our main contributions are summarized as follows.

- We propose a novel out-of-scope intent detection approach by matching training and test tasks to bridge the gap between fitting to training data and generalizing to test data.

- We propose to efficiently construct two types of pseudo outliers by using a simple self-supervised method and leveraging publicly available auxiliary datasets.

- We conduct extensive experiments on four real-world dialogue datasets to demonstrate the effectiveness of our method and perform a detailed ablation study.

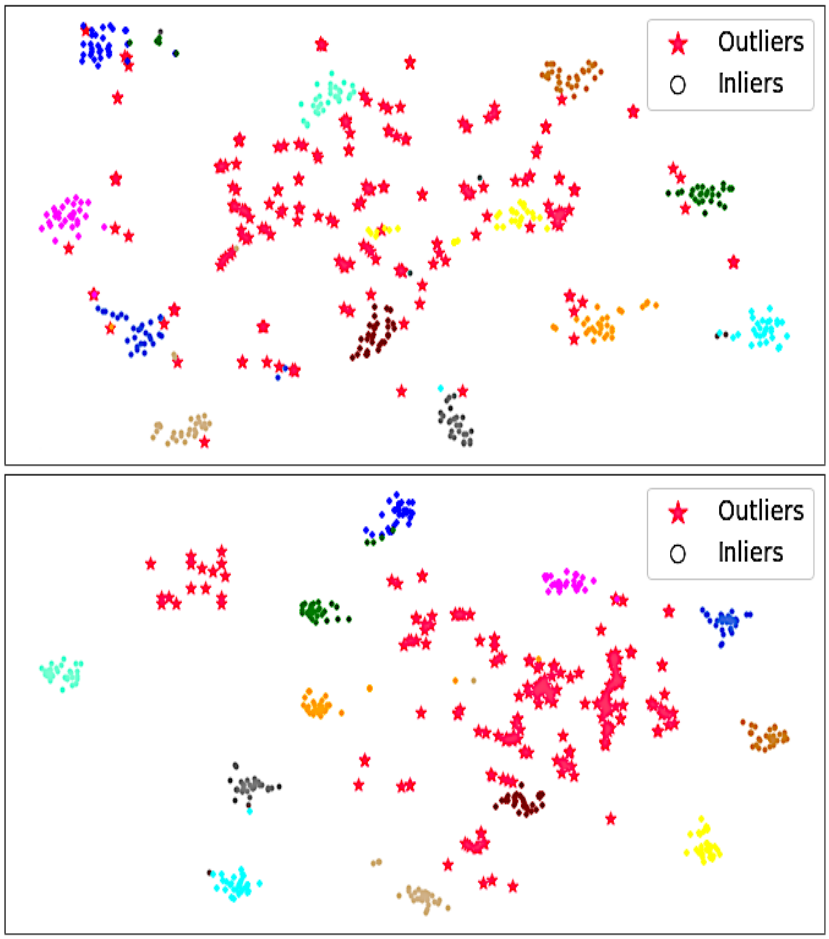

Figure 3.1: t-SNE visualization of the learned embeddings of the test samples of CLINC150. Top: Previous K-way training; Bottom: Our proposed $(K+1)$-way training. Better view in color and enlarged.

3.2 Methodology

Problem Statement In a dialogue system, given K predefined intent classes $S_{\text{known}} = \{C_i\}_{i=1}^{K}$, an unknown intent detection model aims at predicting the category of an utterance u, which may be one of the known intents or an out-of-scope intent C_{oos}. Essentially, it is a $K+1$ classification problem at the test stage. At the training stage, a set of

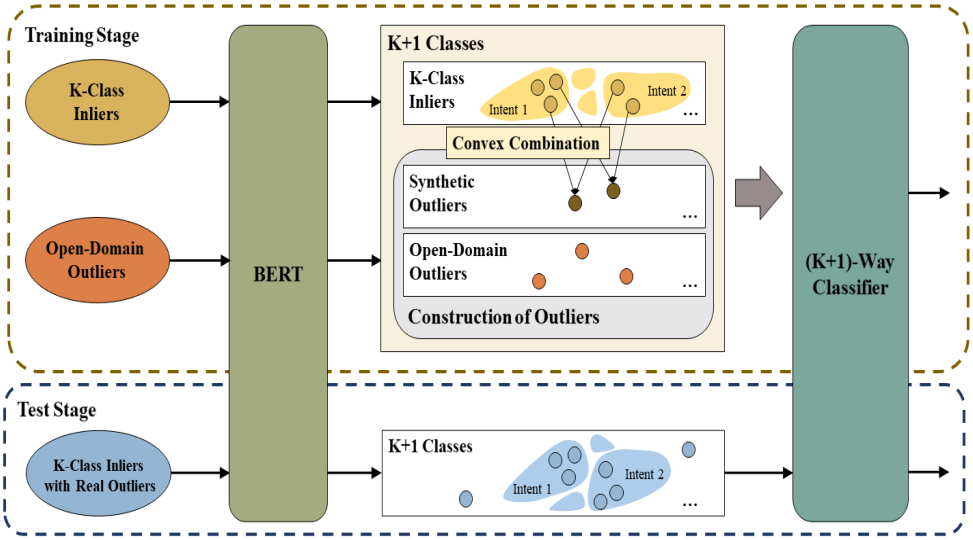

Figure 3.2: An illustration of our proposed method. We use BERT as the utterance encoder. At training stage, we train a (K+1)-way classifier by constructing two types of pseudo outliers. The open-domain outliers are collected from an auxiliary dataset disjoint from both the training and test data. The synthetic self-supervised outliers are generated during training by random convex combinations of features of inliers from different known classes.

N labeled utterances $\mathcal{D}_l = \{(x_i, c_i) \mid c_i \in S_{\text{known}})\}_{i=1}^{N}$ is provided for training. Previous methods typically train a K-way classifier for the known intents.

Overview of Our Approach The mismatch between the training and test tasks, i.e., K-way classification vs. $(K+1)$-way classification, leads to the use of strong assumptions and additional complexity in previous methods. Inspired by recent practice in meta learning to simulate test conditions in training (Vinyals et al., 2016), we propose to match the training and test settings. In essence, as shown in Figure 3.2, we formal-

ize a $(K+1)$-way classification task in the training stage by constructing out-of-scope samples via self-supervision and from open-domain data. Our method simply trains a $(K+1)$-way classifier without making any assumption on the data distribution. After training, the classifier can be readily applied to the test task without any adaptation or post-processing. In the following, we elaborate on the details of our proposed method, including representation learning, construction of pseudo outliers, and discriminative training.

3.2.1 Representation Learning

We employ BERT (Devlin et al., 2019) – a deep Transformer network as text encoder. Specifically, we take the d-dimensional output vector of the special classification token [CLS] as the representation of an utterance u, i.e.,

$$h = \text{BERT}(u) \in \mathbb{R}^d,$$

where $d = 768$ by default. The training set $\mathcal{D}_l$ is then mapped to $\mathcal{D}_l^{tr} = \{(h_i, c_i) \mid h_i = \text{BERT}(u_i), (u_i, c_i) \in \mathcal{D}_l\}_{i=1}^N$ in the feature space.

3.2.2 Construction of Outliers

We construct two different types of pseudo outliers to be used in the training stage: synthetic outliers that are generated by self-supervision,

and open-domain outliers that can be easily acquired.

Synthetic Outliers by Self-Supervision To improve the generalization ability of the unknown intent detection model, we propose to generate "*hard*" outliers in the feature space, which may have similar representations to the inliers of known intent classes. We hypothesize that those outliers may be geometrically *close* to the inliers in the feature space. Based on this assumption, we propose a self-supervised method to generate the "hard" outliers using the training set $\mathcal{D}_l^{tr}$.

Specifically, in the feature space, we generate synthetic outliers by using convex combinations of the features of inliers from different intent classes:

$$h^{oos} = \theta * h_\beta + (1 - \theta) * h_\alpha, \tag{3.1}$$

where h_β and h_α are the representations of two utterances which are randomly sampled from different intent classes in $\mathcal{D}_l^{tr}$, i.e., $c_\beta \neq c_\alpha$, and h^{oos} is the synthetic outlier. For example, θ can be sampled from a uniform distribution $U(0, 1)$. In this case, when θ is close to 0 or 1, it will generate "harder" outliers that only contain a small proportion of mix-up from different classes. In essence, "hard" outliers act like support vectors in SVM (Cortes and Vapnik, 1995), and "harder" outliers could help to

train a more discriminative classifier.

The generated outliers h^{oos} are assigned to the class of C_{oos}, the $(K+1)$-th class in the feature space, forming a training set

$$\mathcal{D}_{co}^{tr} = \{(h_i^{oos}, c_i = C_{oos})\}_{i=1}^{M}. \tag{3.2}$$

Notice that since the outliers are generated in the feature space, it is very efficient to construct a large outlier set $\mathcal{D}_{co}^{tr}$.

Open-Domain Outliers In practical dialogue systems, user input can be arbitrary free-form sentences. To simulate real-world outliers and provide learning signals representing them in training, we propose to construct a set of open-domain outliers, which can be easily obtained. Specifically, the set of free-form outliers $\mathcal{D}_{fo}$ can be constructed by collecting sentences from various public datasets that are disjoint from the training and test tasks. There are many datasets available, including the question answering dataset SQuaD 2.0 (Rajpurkar et al., 2018), the sentiment analysis datasets Yelp (Meng et al., 2018) and IMDB (Maas et al., 2011), and dialogue datasets from different domains.

In the feature space, $\mathcal{D}_{fo}$ is mapped to $\mathcal{D}_{fo}^{tr} = \{(h_i^{oos}, c_i = C_{oos}) \mid h_i^{oos} = \text{BERT}(u_i), u_i \in \mathcal{D}_{fo}\}_{i=1}^{H}$.

Both synthetic outliers and open-domain outliers are easy to con-

struct. As will be demonstrated in Section 3.3, both of them are useful, but synthetic outliers are much more effective than open-domain outliers in improving the generalization ability of the trained $(K+1)$-way intent classifier.

3.2.3 Discriminative Training

After constructing the pseudo outliers, in the feature space, our training set $\mathcal{D}^{tr}$ now consists of a set of inliers $\mathcal{D}_l^{tr}$ and two sets of outliers $\mathcal{D}_{co}^{tr}$ and $\mathcal{D}_{fo}^{tr}$, i.e., $\mathcal{D}^{tr} = \mathcal{D}_l^{tr} \cup \mathcal{D}_{co}^{tr} \cup \mathcal{D}_{fo}^{tr}$ and $|\mathcal{D}^{tr}| = N + M + H$. Therefore, in the training stage, we can train a $(K+1)$-way classifier with the intent label set $S = S_{known} \cup \{C_{oos}\}$, which can be directly applied in the test stage to identify unknown intent and classify known ones. In particular, we use a multilayer perceptron network, $\Phi(\cdot)$, as the classifier in the feature space. The selection of the classifier is flexible, and the only requirement is that it is differentiable. Then, we train our model using a cross-entropy loss:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}^{tr}|} \sum_{\mathcal{D}^{tr}} \log \frac{\exp(\Phi(h_i)^{c_i}/\tau)}{\sum_{j \in S} \exp(\Phi(h_i)^{j}/\tau)},$$

where $\Phi(h_i)^{c_i}$ refers to the output logit of $\Phi(\cdot)$ for the ground-truth class c_i, and $\tau \in \mathbb{R}^+$ is an adjustable scalar temperature parameter.

3.3 Experiments

		CLINC150		StackOverflow		Banking		M-CID-EN	
	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
25%	MSP	66.60	51.20	33.94	45.68	48.15	48.47	52.05	43.14
	DOC	64.43	44.60	60.68	60.51	37.78	46.35	49.32	46.59
	SEG	72.86	65.44	47.00	52.83	51.11	55.68	44.51	50.14
	LMCL	68.57	62.42	41.60	48.21	52.77	56.73	41.44	46.99
	Softmax	76.50	67.74	46.17	50.78	57.88	58.32	41.95	45.46
	Ours	**88.44**	**80.73**	**68.74**	**65.64**	**74.11**	**69.93**	**87.08**	**79.67**
50%	MSP	68.61	51.20	56.33	62.92	53.83	65.33	61.21	54.33
	DOC	62.46	70.01	61.62	68.97	58.29	57.30	59.97	62.28
	SEG	77.05	79.42	68.50	74.18	68.44	76.48	67.91	72.37
	LMCL	78.63	80.42	64.34	71.80	63.59	73.99	63.42	69.04
	Softmax	82.47	82.86	65.96	71.94	67.44	74.19	64.72	69.35
	Ours	**88.33**	**86.67**	**75.08**	**78.55**	**72.69**	**79.21**	**81.05**	**79.73**
75%	MSP	73.41	81.81	76.73	77.63	71.92	80.77	72.89	77.34
	DOC	74.63	78.63	63.98	62.07	72.02	78.04	69.79	71.18
	SEG	81.92	86.57	80.83	84.78	78.87	85.66	75.73	79.97
	LMCL	84.59	88.21	80.02	84.47	78.66	85.33	77.11	80.96
	Softmax	86.26	89.01	77.41	82.28	78.20	84.31	76.99	80.82
	Ours	**88.08**	**89.43**	**81.71**	**85.85**	**81.07**	**86.98**	**80.24**	**82.75**

Table 3.1: Overall accuracy and macro f1-score for unknown intent detection with different proportion of seen classes. For each setting, the best result is marked in bold.

In this section, we present the experimental results of our proposed method on the targeted task of unknown intent detection. Given a test set comprised of known and unknown intent classes, the primary goal of an unknown intent detection model is to assign correct intent labels to utterances in the test set. Notice that the unknown intent label C_{oos} is also included as a special class for prediction.

3.3.1 Datasets and Baselines

We evaluate our proposed method on four benchmark datasets as follows, three of which are newly released dialogue datasets designed for intent detection. The statistics of the datasets are summarized in Table 3.2.

Dataset	Vocab	Avg. Length	Samples	Classes
CLINC150	8,376	8.31	23,700	150
StackOverflow	17,182	9.18	20,000	20
Banking	5028	11.9	13,083	77
M-CID-EN	1,254	6.74	1,745	16

Table 3.2: Dataset statistics.

		CLINC150		StackOverflow		Banking		M-CID-EN	
	Methods	Unknown	Known	Unknown	Known	Unknown	Known	Unknown	Known
25%	MSP	73.20	50.62	22.59	50.30	49.98	48.39	56.27	37.86
	DOC	71.08	43.91	66.11	59.39	31.41	47.14	53.08	44.92
	SEG	79.90	65.06	46.17	54.16	53.22	55.81	42.73	51.99
	LMCL	75.61	62.01	38.85	50.15	55.29	56.81	36.99	49.50
	Softmax	83.04	67.34	45.52	51.83	62.52	58.10	35.39	46.22
	Ours	**92.35**	**80.43**	**74.86**	**63.80**	**80.12**	**69.39**	**91.15**	**76.80**
50%	MSP	57.78	68.03	35.18	70.09	29.31	66.28	58.55	53.80
	DOC	57.62	70.17	47.96	71.07	49.88	57.50	47.22	64.16
	SEG	78.02	79.43	60.89	75.51	60.42	76.90	61.04	73.80
	LMCL	79.89	80.42	53.12	71.80	50.30	74.62	51.11	71.29
	Softmax	84.19	82.84	56.80	73.45	60.28	74.56	56.30	70.98
	Ours	**90.30**	**86.54**	**71.88**	**79.22**	**67.26**	**79.52**	**82.44**	**79.39**
75%	MSP	57.83	82.02	41.73	80.03	23.86	81.75	39.56	80.50
	DOC	64.62	78.76	49.50	62.91	39.47	78.72	49.41	72.99
	SEG	76.12	86.67	62.30	86.28	54.43	86.20	51.51	82.34
	LMCL	80.42	88.28	61.40	84.47	53.26	85.89	54.61	83.16
	Softmax	83.12	**89.61**	54.07	84.11	56.90	84.78	58.73	82.66
	Ours	**86.28**	89.46	**65.44**	**87.22**	**60.71**	**87.47**	**69.00**	**83.89**

Table 3.3: Macro f1-score of the known classes and f1-score of the unknown class with different proportion of seen classes. For each setting, the best result is marked in bold.

CLINC150 (Larson et al., 2019) is a dataset specially designed for out-of-scope intent detection, which consists of 150 known intent classes from 10 domains. The dataset includes $22,500$ in-scope queries and $1,200$ out-of-scope queries. For the in-scope ones, we follow the original splitting, i.e., $15,000$, $3,000$ and $4,500$ for training, validation, and testing respectively. For the out-of-scope ones, we group all of the $1,200$ queries into the test set.

StackOverflow (Xu et al., 2015) consists of 20 classes with $1,000$

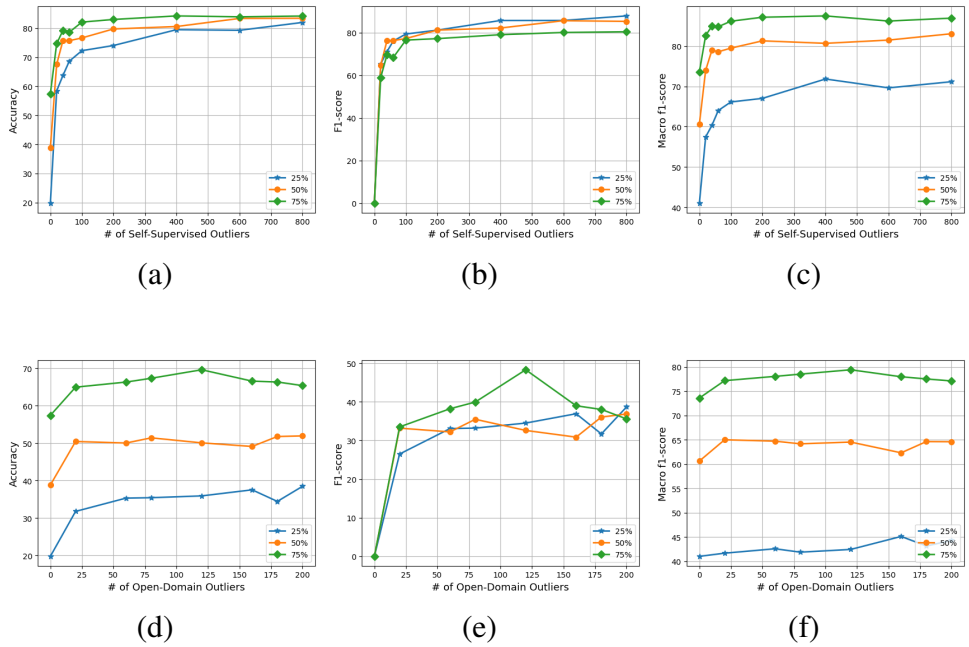

Figure 3.3: Effect of the number of pseudo outliers on CLINC150. (a), (b), and (c) display overall accuracy, f1-score on the unknown class and overall macro f1-score with varying number of self-supervised outliers respectively. (d), (e), and (f) display the corresponding results with varying number of open-domain outliers.

examples in each class. We follow the original splitting, i.e., $12,000$ for training, $2,000$ for validation, and $6,000$ for test.

Banking (Casanueva et al., 2020) is a fine-grained intent detection dataset in the banking domain. It consists of $9,003$, $1,000$, and $3,080$ user queries in the training, validation, and test sets respectively.

M-CID (Arora et al., 2020) is a recently released dataset related to Covid-19. We use the English subset of this dataset referred to as M-CID-EN in our experiments, which covers 16 intent classes. The splitting of M-CID-EN is $1,258$ for training, 148 for validation, and 339

for test.

We extensively compare our method with the following unknown intent detection methods.

- **Maximum Softmax Probability (MSP)** (Hendrycks and Gimpel, 2017) employs the confidence score derived from the maximum softmax probability to predict the class of a sample. The idea under the hood is that the lower the confidence score is, the more likely the sample is of an unknown intent class.

- **DOC** (Shu et al., 2017) considers to construct m 1-vs-rest sigmoid classifiers for m seen classes respectively. It uses the maximum probability from these classifiers as the confidence score to conduct classification.

- **SEG** (Yan et al., 2020) models the intent distribution as a margin-constrained Gaussian mixture distribution and uses an additional outlier detector – local outlier factor (LOF) (Breunig et al., 2000) to achieve unknown intent detection.

- **LMCL** (Lin and Xu, 2019) considers to learn discriminative embeddings with a large margin cosine loss. It also uses LOF as the outlier detection algorithm.

- **Softmax** (Lin and Xu, 2019) uses a softmax loss to learn discrimi-

native features based on the training dataset, which also requires an additional outlier detector such as LOF for detecting the unknown intents.

3.3.2 Experimental Setup and Evaluation Metrics

To compare with existing methods, we follow the setting in LMCL (Lin and Xu, 2019). Specifically, for each dataset, we randomly sample 75%, 50%, and 25% of the intent classes from the training set as the known classes to conduct training, and we set aside the rest as the unknown classes for test. Notice that for training and validation, we only use data within the chosen known classes and do not expose our model to any of test-time outliers. Unless otherwise specified, in each training batch, we keep the ratio of inliers, open-domain outliers and self-supervised outliers roughly as $1 : 1 : 4$. This setting is empirically chosen and affected by the memory limit of NVIDIA 2080TI GPU, which we use for conducting the experiments. The number of pseudo outliers can be adjusted according to different environments, and a larger number of self-supervised outliers typically takes more time to converge.

We use Pytorch (Paszke et al., 2019) as the backend to conduct the experiments. We use the pretrained BERT mdoel (*bert-base-uncased*) provided by Wolf et al. (2019) as the encoder for utterances. We use the

output vector of the special classification token [CLS] as the utterance embedding and fix its dimension as 768 by default throughout all of our experiments. To ensure a fair comparison, all baselines and our model use the same encoder.

For model optimization, we use AdamW provided by Wolf et al. (2019) to fine-tune BERT and Adam proposed by Kingma and Ba (2015) to train the MLP clasisfier $\Phi(\cdot)$. We set the learning rate for BERT as $1e^{-5}$ as suggested by Devlin et al. (2019). For the MLP clasisfier, the learning rate is fixed as $1e^{-4}$. Notice that the fine-tuning of BERT is conducted simultaneously with the training of the classifier $\Phi(\cdot)$ with the same cross-entropy loss. The MLP classifier $\Phi(\cdot)$ has a two-layer architecture with $[1024, 1024]$ as hidden units. The temperature parameter τ is selected by cross-validation and set as 0.1 in all experiments.

Following LMCL (Lin and Xu, 2019), we use overall accuracy and macro f1-score as evaluation metrics. All results reported in this section are the average of 10 runs with different random seeds, and each run is stopped until reaching a plateau on the validation set. For baselines, we follow their original training settings except using the aforementioned BERT as text encoder.

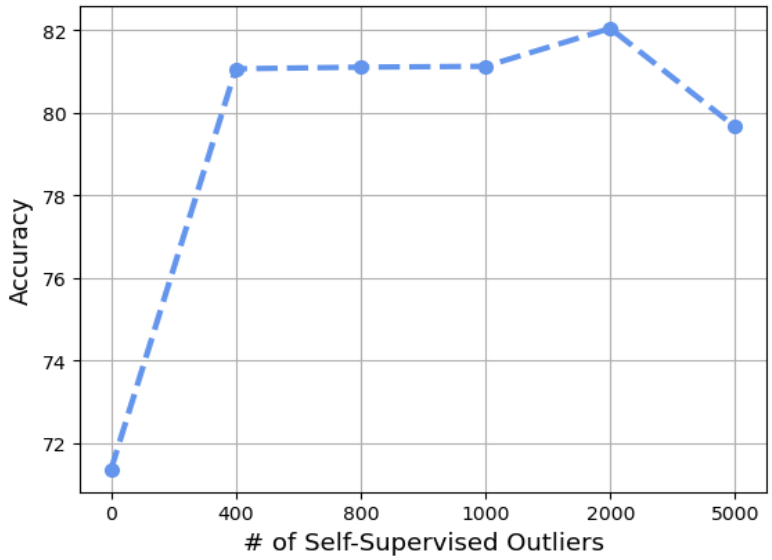

Figure 3.4: Effect of the number of self-supervised outliers on overall intent detection accuracy under the 75% setting of Banking.

3.3.3 Result Analysis

We present our main results in Table 3.1 and Table 3.3. Specifically, Table 3.1 gives results in overall accuracy and macro f1-score for all classes including the outlier class, while Table 3.3 shows results in macro f1-score for the known classes and f1-score for the outlier class respectively. It can be seen that, on all benchmarks and in almost every setting, our model significantly outperforms the baselines. As shown in Table 3.3, our method achieves favorable performance on both unknown and known intent classes simultaneously.

It is worth mentioning that the large improvements of our method in scenarios with small labeled training sets (25% and 50% settings)

indicate its great potential in real-life applications, since a practical dialogue system often needs to deal with a larger proportion of outliers than inliers due to different user demographic, ignorance/unfamiliarity of/with the platform, and limited intent classes recognized by the system (especially at the early development stage).

More importantly, referring to Table 3.3, as the proportion of known intents increases, it can be seen that the performance gains of the baselines mainly lie in the known classes. In contrast, our method can strike a better balance between the known and unknown classes without relying on additional outlier detector, margin tuning, and threshold selection, demonstrating its high effectiveness and generality. Take the Softmax baseline for example, in the 75% case of CLINC150, it achieves a slightly higher result than our model on the known classes but a substantially lower result on the unknown ones.

3.3.4 Effect of Pseudo Outliers

We conduct an ablation study on the effectiveness of the two kinds of pseudo outliers and summarize the results in Table 3.4. The first row of the three settings (25%, 50%, and 75%) stands for training solely with the labeled examples of CLINC150 without using any pseudo outliers. In general, self-supervised synthetic outliers and open-domain outliers

both lead to positive effects on classification performance. For each setting, comparing the second row with the third, we can observe that the synthetic outliers produced by convex combinations lead to a much larger performance gain than that of pre-collected open-domain outliers. Finally, combining them for training leads to the best results, as shown in the fourth row of each setting.

Next, we conduct experiments to study the impact of varying the number of the two kinds of pseudo outliers separately, as shown in Figure 3.3. We first fix the number of open-domain outliers as zero and then increase the number of self-supervised outliers. The results are displayed in Figure 3.3 (a), (b) and (c). In particular, as the number of self-supervised outliers grows, the performance first increases quickly and then grows slowly. On the other hand, we fix the number of self-supervised outliers as zero and then increases the number of open-domain outliers. The results are shown in Figure 3.3 (d), (e) and (f), where it can be seen that dozens of open-domain outliers already can bring significant improvements, though the gain is much smaller compared to that of the self-supervised outliers.

Finally, we investigate the impact of the number of self-supervised outliers on overall intent detection accuracy with both the number of inliers and the number of open-domain outliers fixed as 100 per training

batch. As shown in Figure 3.4, we increase the number of self-supervised outliers from 0 to 5000. Note that 400 is the default setting used in Table 3.1 and Table 3.3. We can see that comparable results can be obtained for a wide range of numbers. However, when the number grows to 5000, the performance exhibits a significant drop. We hypothesize that as the number increases, the generated synthetic outliers may be less accurate, because some convex combinations may fall within the scope of known classes.

To summarize, self-supervised outliers play a much more important role than open-domain outliers for unknown intent classification. Self-supervised outliers not only provide better learning signals for the unknown intents, but also impose an important positive effect on the known ones. For the open-domain outliers, if used alone, they can only provide limited benefit. But in combination with the self-supervised ones, they can further enhance the performance.

3.3.5 Selection of Open-Domain Outliers

To demonstrate the flexibility of our method in selecting open-domain outliers as described in Section 3.2.2, we train our model on CLINC150 using open-domain outliers from different sources. The results are summarized in Table 3.5. Specifically, Open-bank and Open-stack stand

	$\mathcal{D}_{co}^{tr}$	$\mathcal{D}_{fo}^{tr}$	Acc	Macro-F1	F1 Unknown
25%			19.79	41.05	-
	✓		81.96	71.15	87.8
		✓	37.55	45.14	36.91
	✓	✓	88.44	80.73	92.35
50%			38.78	60.35	-
	✓		83.12	82.62	85.03
		✓	48.62	63.19	28.82
	✓	✓	88.33	86.67	90.30
75%			57.43	73.6	-
	✓		84.16	86.9	80.36
		✓	69.61	79.42	48.29
	✓	✓	88.08	89.43	86.28

Table 3.4: An ablation study on the effectiveness of pseudo outliers.

	$\mathcal{D}_{fo}^{tr}$	Acc	Macro-F1
25%	Open-bank	89.36	81.22
	Open-stack	88.38	80.42
	Open-big	88.44	80.73
50%	Open-bank	87.35	86.41
	Open-stack	88.23	86.37
	Open-big	88.33	86.67
75%	Open-bank	87.19	89.33
	Open-stack	87.52	89.17
	Open-big	88.08	89.43

Table 3.5: Results on CLINC150 with different sets of open-domain outliers.

for using the training set of Banking and StackOverflow as the source of open-domain outliers respectively. Open-big stands for the source of open-domain outliers used in other experiments, which consists of ~ 0.5 million sentences randomly selected from SQuaD 2.0 (Rajpurkar et al., 2018), Yelp (Meng et al., 2018), and IMDB (Maas et al., 2011). It can

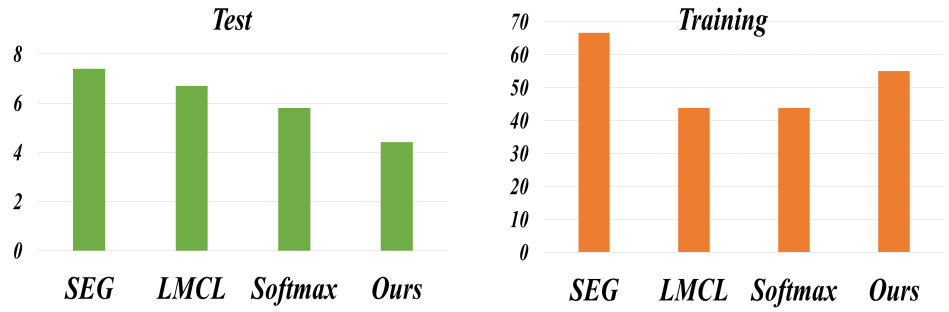

Figure 3.5: Comparison of training time (per epoch) and test time with baselines.

be seen that the performance of our model is insensitive to the selection of open-domain outliers.

3.3.6 Efficiency

We provide a quantitative comparison on the training and test efficiency for our method and the baselines, by calculating the average time (in seconds) for training per epoch and the total time for testing under the 75% setting. Here, we only compare with the strongest baselines. As shown in Figure 3.5, even with the pseudo outliers, the training time of our method is comparable to that of the baselines. Importantly, in the test stage, our method demonstrates significant advantages in efficiency, which needs much less time to predict intent classes for all samples in the test set.

3.4 Chapter Review

In this chapter, we have proposed a simple, effective, and efficient approach for out-of-scope intent detection by overcoming the limitation of previous methods via matching train-test conditions. Particularly, at the training stage, we construct self-supervised and open-domain outliers to improve model generalization and simulate real outliers in the test stage. Extensive experiments on four dialogue datasets show that our approach significantly outperforms state-of-the-art methods. In the future, we plan to investigate the theoretical underpinnings of our approach and apply it to more applications.

Chapter 4

Low-resource OOD Detection

4.1 Introduction

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

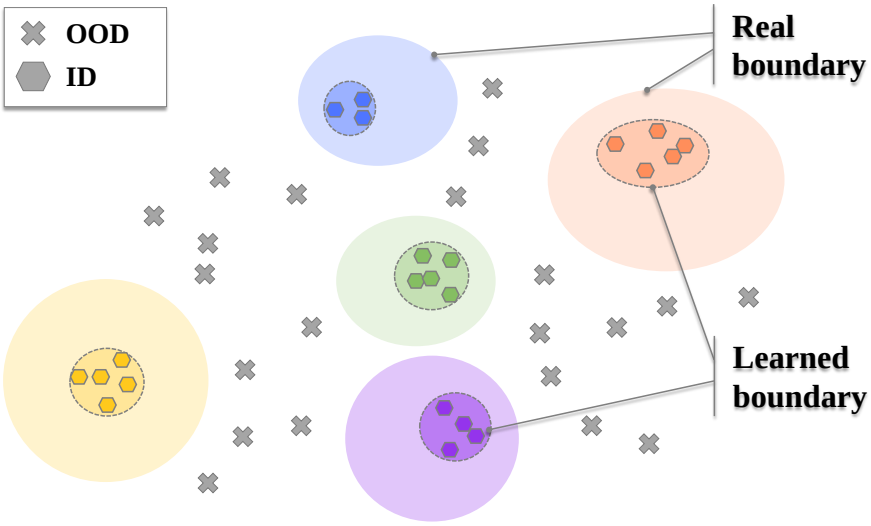

Figure 4.1: The challenge of few-shot out-of-distribution intent detection. OOD stands for out-of-distribution examples and ID stands for in-distribution examples.

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

In this work, we take a closer look at few-shot OOD intent detection and consider a strict setting, where only few-shot in-distribution labeled

examples are available during training and no external resources can be exploited, since the requirement of additional resources hinders the applicability of the model. Under this simplified yet more challenging setting, state-of-the-art OOD intent detection algorithms fail to achieve acceptable performance. In Figure 4.1, we illustrate the key challenge for few-shot OOD detection. As shown in Figure 4.1, since ID classes are under-represented by few-shot ID examples, a model based on density estimation (Zhang et al., 2021) or $(K + 1)$-way discriminative training (Zhan et al., 2021) tends to learn a conservative decision boundary and hence there are large margins between the real and learned decision boundaries. Real ID examples situate in the margins will be inaccurately assigned to the OOD class, leading to poor performance.

Therefore, the key for few-shot OOD intent detection is to improve the model performance on ID examples. To address this issue, we propose to enrich the training set to improve the representativeness of ID intent classes and provide more useful learning signals. We explore the feasibility of generating synthetic ID examples in a self-supervised manner. In particular, we train a denoising autoencoder (DAE) (Vincent et al., 2008) in the latent representation space only using the few labeled ID examples. The trained decoder of DAE is then used to efficiently sample synthetic ID examples. With the enlarged training set, we train

a $(K + 1)$-way classifier proposed in Chapter 3 by simulating OOD examples. Our contributions are summarized as follows:

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

4.2 Pilot Study

Out-of-distribution (OOD) intent detection aims at improving the robustness of a dialogue system with respect to utterances with unknown (or unsupported) intents. The key challenge of OOD detection is that real OOD samples are inaccessible during training and validation. Given an in-distribution (ID) set of K known classes, $y_i \in \{y^k\}_{k=1}^{K}$, the OOD detection task considers another special OOD class y^{OOD} to represent

any malformed or unsupported utterances. Hence, given the input space $\mathcal{X} \times \mathcal{Y}$, the goal of OOD intent detection is to learn a $(K+1)$-way classifier $f_\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ to minimize the expected risk:

$$R(f) = \mathbb{E}(1[f_\phi(x_i) \neq y_i)]), \tag{4.1}$$

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

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

Pilot study. To illustrate the challenges of few-shot OOD intent detection, we conduct a pilot study on a commonly used OOD intent detection dataset CLINC150 (Larson et al., 2019) using two recent state-of-the-art approaches (Zhang et al., 2021; Zhan et al., 2021) for few-shot OOD intent detection. To simulate the few-shot scenario, in the experiment, only 5 labeled examples in each ID class are used for

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

Table 4.1: A pilot study on few-shot OOD intent detection. DCL (Zhan et al., 2021) and ADB (Zhang et al., 2021) are two recent state-of-the-art approaches for OOD intent detection. ID-F1 indicates macro f1-score on the in-distribution classes. OOD-F1 stands for f1-score on the out-of-distribution class.

training. The results are summarized in Table 4.1. For OOD detection, we randomly select 25%, 50% and 75% intent classes as in-domain classes and assign the remaining classes to the OOD category. Experimental details are elaborated in Section 3.3.

We can observe that both of the two methods yield unsatisfactory performance. Specifically, the performance on the ID classes is poor and way lower than that on the OOD class. When there are only 25% ID classes (~ 38), the gap between the ID and OOD classes in f1-score is the largest (up to $44+$). Although moderate overall accuracy is achieved, such OOD intent detection model can only provide services to users worse than random choices, since the majority of user utterances are rejected as OOD inputs. It also indicates that the overall accuracy may not be a good performance measure for this task. These observations show

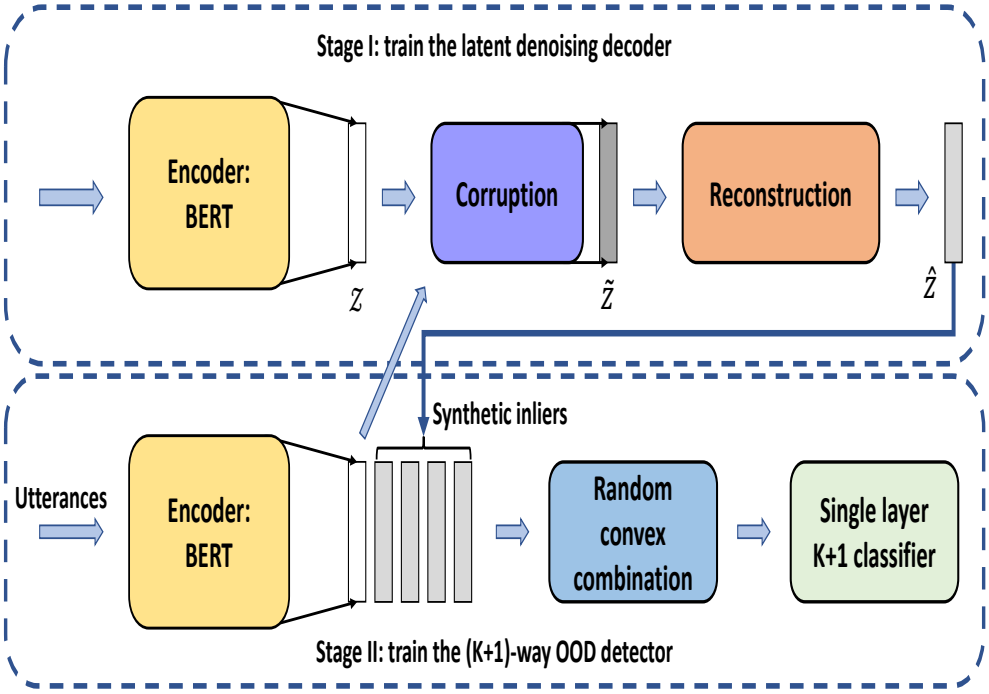

Figure 4.2: An overview of our proposed framework.

that in the few-shot scenario, existing OOD intent detection algorithms can be easily biased towards the OOD class, due to inadequate representations of the ID classes. Hence, directly applying them to few-shot OOD intent detection will lead to sub par performance.

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

4.3 Methodology

4.3.1 Utterance Representation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training set, where x_i denotes an input token sequence with size m, i.e., $[x_i^0, \cdots, x_i^{m-1}]$. For each input x_i, we use BERT as the encoder to map x_i into a sequence of hidden states h_i, i.e., BERT:$\mathcal{X} \rightarrow \mathcal{H}$ and $h_i \in \mathbb{R}^{(m+1)*768}$. Note that for every sentence, BERT adds a spacial token [CLS] at the beginning of the sequence. Following common practice, we use the average pooling of the hidden sequence h_i as the representation of an utterance:

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

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

4.3.2 Our Proposed Model

As shown in Figure 4.2, we propose a two-stage model for few-shot OOD intent detection. In the first stage, we learn a stochastic reconstruction function to generate synthetic ID samples in the representation space to enrich the in-distribution training set. In the second stage, we adopt a

$(K+1)$-way discriminative training procedure for OOD detection by simulating OOD examples based on the enlarged in-distribution training set. Notice that throughout the two stages, we only use the few labeled in-distribution data without exploiting external labeled intent detection data or fine-tuning corpus. Our algorithm is summarized in Algorithm (4.1).

Stage I: Generating Synthetic In-distribution Data

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

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

To this end, we employ a principled distribution estimation method – denoising autoencoder (DAE) – to build an efficient stochastic process for

sampling ID examples with a consistency guaranteed estimator for $p(z)$. The latent DAE consists of two components: the corruption distribution $\mathcal{C}(\tilde{z} \mid z)$ and the reconstruction distribution $q_\theta(z \mid \tilde{z})$. The DAE can be learned by:

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

where the likelihood is computed by a mean square loss between the original embedding vector z and the reconstructed vector $\hat{z}$ as shown in Figure 4.2.

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

$$\begin{aligned} \hat{z} &\sim q_{\theta*}(z \mid \tilde{z}), \\ \tilde{z} &\sim \mathcal{C}(\tilde{z} \mid z). \end{aligned} \tag{4.2}$$

The corruption distribution $\mathcal{C}$ can be instantiated by simple stochastic operations like Dropout Srivastava et al. (2014). By repeatedly applying the process in Equation (4.2), we can obtain a synthetic labeled ID set $\mathcal{D}^{\text{rec}} = \{(\hat{z}_i, y_i)\}_{i=1}^{L}$, where the reconstructed representation $\hat{z}_i$ shares the same label y_i with the original uncorrupted z_i. Finally, by combining the original training set $\mathcal{D}^{tr}$ and the synthetic set $\mathcal{D}^{\text{rec}}$, we get an enlarged labeled training set $\mathcal{D}^{\text{Enlarged}} = \mathcal{D}^{tr} \cup \mathcal{D}^{\text{rec}}$.

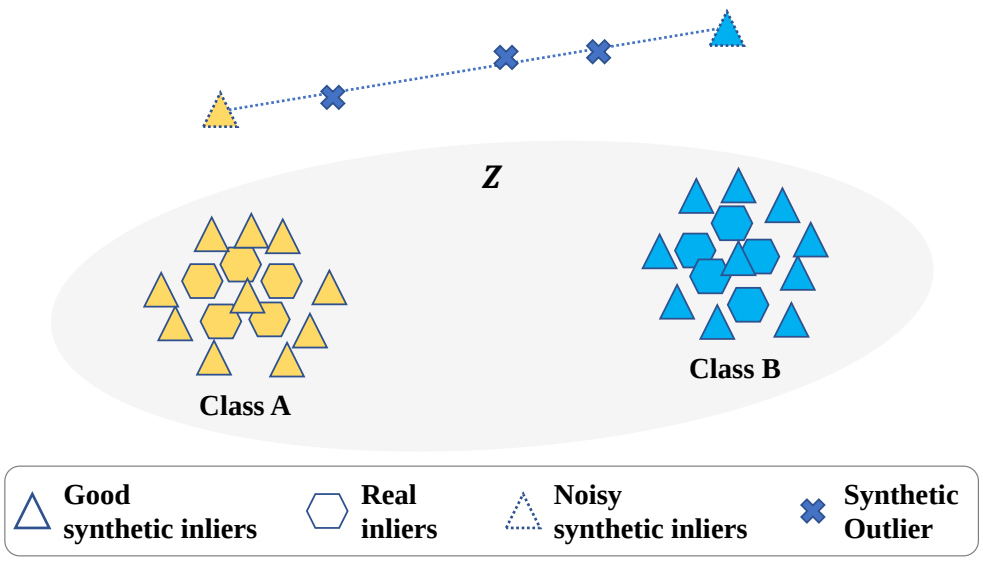

Figure 4.3: Illustration of the noise neutralizing effect under the $(k+1)$-way training paradigm.

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

As shown in the Figure 4.2, the second stage of our proposed method aims at learning a $(K+1)$-way classifier in an end-to-end manner. Since only few-shot samples are used to train the reconstruction distribution $q_\theta(z \mid \tilde{z})$, the resulting $q_{\theta*}(\cdot)$ may not be a perfect estimator for the true distribution, and the enlarged in-distribution set $\mathcal{D}^{\text{Enlarged}}$ may be noisy. Hence, it may not be the best choice to directly apply density estimation-based methods for OOD intent detection, due to the risk of overfitting. To better utilize the enlarged in-distribution set $\mathcal{D}^{\text{Enlarged}}$, we adopt the $(K+1)$-way discriminative training strategy proposed in Zhan et al. (2021) and follow their idea to construct OOD learning signals via random convex combination between representations from different

Algorithm 4.1 Our proposed algorithm

Stage I: DAE training: Denoising autoencoder q_θ, few-shot training set $\mathcal{D}^{tr} = \{(z_i, y_i)\}_{i=1}^{N}$, Dropout rate β, learning rate λ_{DAE}.

for all $z_i \in \mathcal{D}^{tr}$ **do**

Corruption and reconstruction:

- $\tilde{z}_i \sim \mathcal{C}(\tilde{z}_i \mid z_i; \beta)$
- $\hat{z}_i \sim q_\theta(z_i \mid \tilde{z}_i)$

Update $\theta^* = \theta - \lambda_{DAE} \nabla_\theta \sum_{i=1}^{N} \| z_i - \hat{z}_i \|_2^2$.

q_{θ^*}.

Stage II: $(k+1)$-way training

Classifier f_ϕ, inlier sampling number N_{ID}, DAE q_{θ^*}, outlier sampling number N_{OOD}

Sample N_{ID} times via the Equation (4.2) $\rightarrow \mathcal{D}^{\text{rec}}$

$\mathcal{D}^{\text{Enlarged}} = \mathcal{D}^{tr} \cup \mathcal{D}^{\text{rec}}$.

N_{OOD} Simulated OOD example construction:

- $(z_i, y_i),\ (z_j, y_j) \sim \mathcal{D}^{\text{Enlarged}}$, $y_i \neq y_j$.
- $\alpha \sim U(0, 1)$.
- $z_i^{OOD} = \alpha * z_i + (1 - \alpha) * z_j$.
- $z_i^{OOD} \rightarrow \mathcal{D}^{\text{OOD}}$.

Minimize the empirical classification risk on $\mathcal{D}^{\text{Enlarged}} \cup \mathcal{D}^{\text{OOD}}$ via the Equation (4.1).

f_{ϕ^*}.

in-distribution classes in the enlarged in-distribution set. By doing so, the impact of noisy synthetic in-distribution examples can be mitigated. We demonstrate this phenomenon in Figure 4.3. The linear interpolation between off-manifold noisy synthetic in-distribution examples tends to represent the OOD examples, since the word embeddings of BERT has been found concentrating near a low-dimensional manifold of the representation space (Ethayarajh, 2019).

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

an OOD set $\mathcal{D}^{\text{OOD}}$ by:

$$z_i^{OOD} = \alpha * z_i + (1 - \alpha) * z_j, \tag{4.3}$$

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

Finally, our $(K + 1)$-way classifier can be learned by minmizing the loss in Equation (4.1) on the union set $\mathcal{D}^{\text{OOD}} \cup \mathcal{D}^{\text{Enlarged}}$.

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

Table 4.2: Dataset statistics.

4.4 Experiments

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

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

Table 4.3: Overall macro f1-score including the OOD class for few-shot OOD intent detection with different proportion (0.25, 0.5 and 0.75) of in-distribution classes. *p* indicates the ratio of selected few-shot in-distribution examples. For each setting, the best result is marked in bold.

4.4.1 Datasets and Baselines

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

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

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

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

The dataset statistics are summarised in Table 4.2.

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

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

- **DOC** (Shu et al., 2017): It shares a similar idea with MSP in assuming that in-distribution examples tend to have higher probability scores. It uses the maximum probability from m 1-vs-rest sigmoid classifiers for m ID classes respectively as the confidence score.

- **LMCL** (Lin and Xu, 2019): It leverages local outlier factor(LOF) to identity samples which are far away from the clusters in the

embedding space as outliers. The model learns discriminative features by largin margin cosine loss.

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

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

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

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

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

Table 4.4: Macro f1-score excluding the OOD class for few-shot OOD intent detection with different proportion (0.25, 0.5 and 0.75) of in-distribution classes. *p* indicates the ratio of selected few-shot in-distribution examples. For each setting, the best result is marked in bold.

4.4.2 Experimental Setup

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

To construct few-shot OOD intent detection tasks from the three datasets, we randomly sample 5% and 10% labeled examples per class as the training set from each of the three datasets. Then, we randomly select 25%, 50%, 75% of the classes in each dataset as in-distribution

(ID) classes and set aside the respective remaining classes to the OOD class for the test stage. Concrete numbers of ID examples per class for each dataset can be found in Table 4.2. In particular, during training and validation, only the labeled few-shot examples of ID classes are seen by the model.

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

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

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

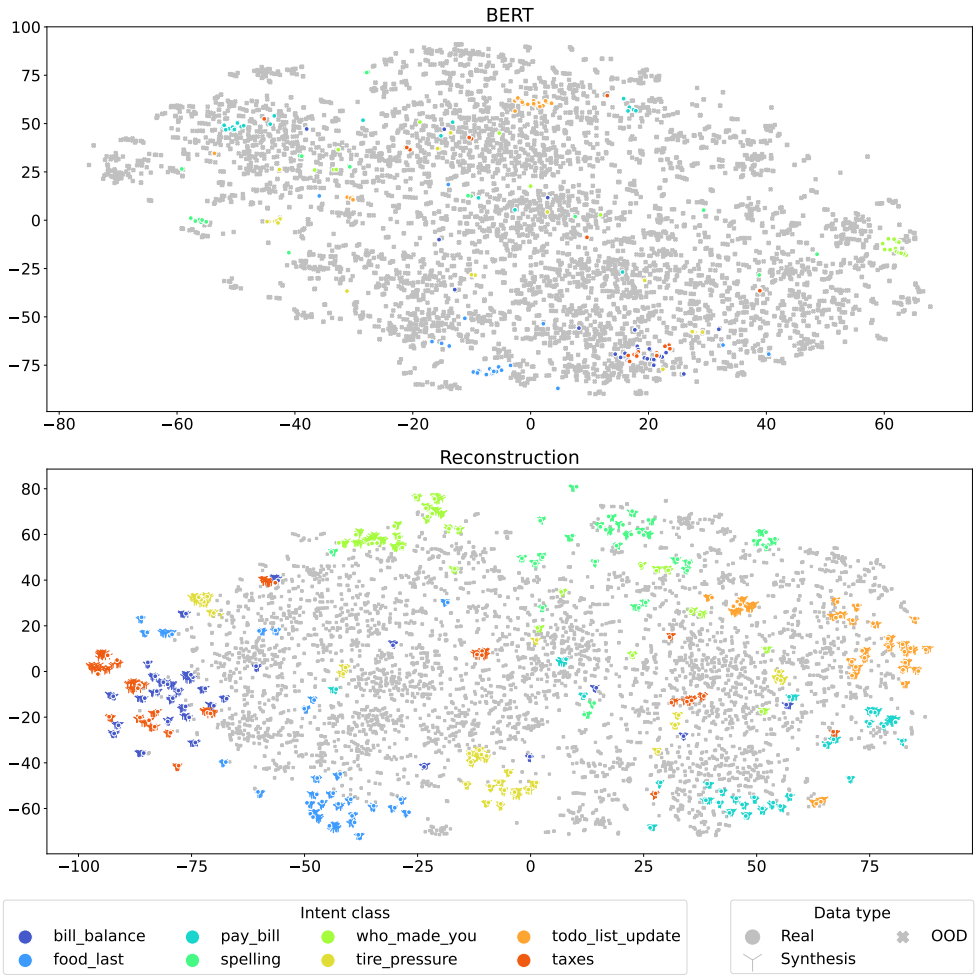

Figure 4.4: t-SNE visualization of BERT embeddings. Top: BERT embeddings without the synthetic in-distribution examples; Bottom: BERT embeddings with the synthetic in-distribution examples. Better view in color and enlarged.

4.4.3 Correctness of the Synthetic In-distribution Examples

In Figure 4.4, we provide a qualitative evaluation of the generated synthetic in-distribution (ID) examples using t-SNE visualization (Van der Maaten and Hinton, 2008). We use the BERT embeddings of 5% labeled examples of 8 ID classes and all out-of-distribution examples from

CLINC150 and plot them on the top of the figure. By generating 10 synthetic ID examples for each real ID example, we have the bottom figure where we can observe that these synthetic ID examples closely situate in the vicinity of each real ID example. Since BERT embeddings have been proved to be rich in contextualized semantics (Devlin et al., 2019), the distance between different embeddings can reflect the semantic gap between them. In this regard, at a high level, our generated ID examples can capture the expressiveness of ID classes.

4.4.4 Main Results

We present the results for the aforementioned three datasets in Table 4.3 and Table 4.4. As shown in the two tables, our proposed method consistently outperforms all baselines by a large margin in all settings.

Table 4.3 presents the results in overall macro f1-score on $(K + 1)$ classes including the OOD class. The results in this table can be interpreted as the overall performance of the model. We first inspect the challenging case, where only 5% labeled examples per class are sampled for training as shown in the top of Table 4.3. We can observe that our method leads to large improvements on all three datasets. In the most challenging case (only 25% of classes in each dataset are selected as in-distribution classes), the improvement is more than 9% on CLINC150

and 8% on Banking than the second best results. Moreover, in the 50% and 75% cases, the improvements are also significant. For example, in the 50% case of Banking, the gap between our method and the second best one is around 6.6%. These results verify the effectiveness and consistency of our model in extreme data-scarcity scenarios. As the ratio of labeled examples per class increased to $p = 10\%$, it can be seen that the baselines are improved by a large margin compared with the case of $p = 5\%$. However, our method can still achieve consistent improvement. This validates the robustness of our method under various data-scarcity scenarios.

In Table 4.4, we summarize the results in macro f1-score of in-distribution classes to demonstrate the effectiveness of synthetic ID examples in our method. It can be seen that in all settings, the performance gains are consistent with the results in Table 4.3, which indicates that the synthetic ID examples sampled from the DAE can help to improve the classification performance on ID classes.

4.4.5 Effectiveness of the Synthetic In-distribution Examples

First, we study the impact of the number of synthetic in-distribution (ID) examples. We conduct experiments on the 5% labeled ratio case.

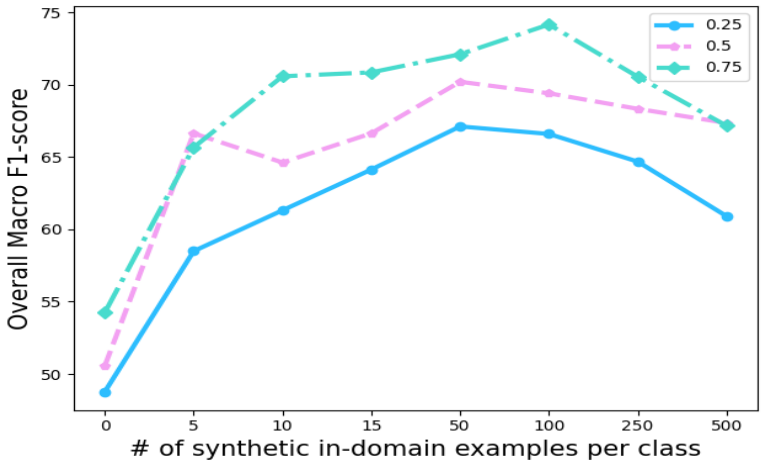

Figure 4.5: Effect of the number of synthetic in-distribution examples.

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

Table 4.5: Results of SEG (Yan et al., 2020) and SEG with our synthetic ID examples (SEG + Ours). ID-F1 stands for in-distribution f1-score, and overall-F1 indicates the macro f1-score for all classes including the OOD class. Better results are marked in bold.

As shown in Figure 4.5, we vary the number of synthetic ID examples per class from 0 to 500. In the range of [0,100], the classification performance increases gradually for all cases (0.25, 0.5 and 0.75). It shows the expressiveness of the synthetic ID examples. However, in the range of [100,500], we observe a slow performance drop in all cases. This is probably because the ID generator is learned from few-shot data and

may generate inaccurate ID examples.

To further verify the effectiveness of our synthetic generator, we incorporate the synthetic ID examples to a strong baseline SEG (Yan et al., 2020) and present the results under the $p = 5\%$ setting of CLINC150 in Table 4.5. With our enlarged ID training set, the performance of SEG can also be improved significantly.

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

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

As shown in Figure 4.6, we vary the corruption rate (from 0% to 100%) of the learned latent denoising autoencoder (DAE) (trained by 30% corruption rate). Notice that 100% corruption rate indicates that no useful reconstruction information is passed to the DAE. We can observe that in the 0.5 (orange line) and 0.75 (green line) cases, the learned $(K+1)$-way classifier can maintain a surprisingly consistent performance compared with the 0.25 (purple line). Especially, with 90% corruption rate, the synthetic in-distribution (ID) examples are much less accurate than those with 30% or 40% corruption rate, but the performance does not drop to an unacceptable level. This verifies the noise neutralization effect of the

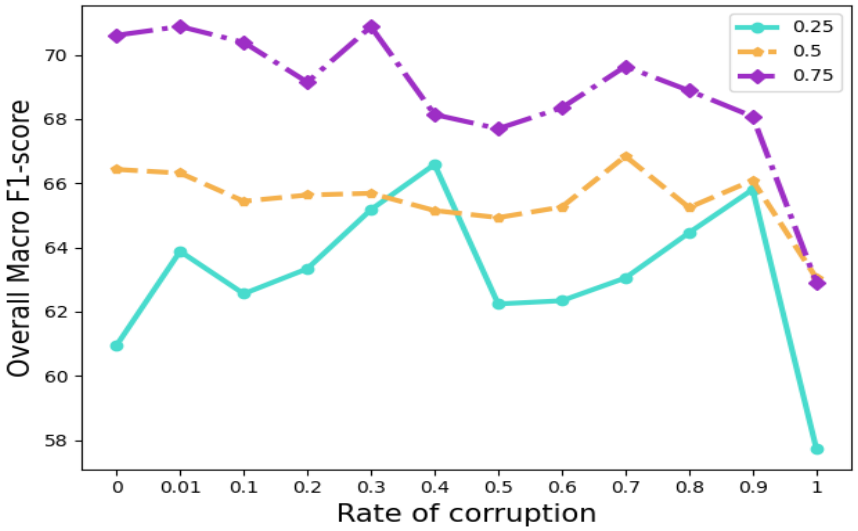

Figure 4.6: Effect of the rate of corruption on the learned denoising autoencoder. The experiment is conducted on CLINC150 under the $p = 5\%$ setting.

$(K + 1)$-way training manner discussed in Section 4.3.

4.5 Chapter Review

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

Chapter 5

A Unified Probabilistic Framework

5.1 Introduction

Large-scale deep neural networks (DNNs) like CNN and Transformers have revolutionized many challenging real-world machine learning applications. However, DNNs still have a significant limitation of making *overconfident* decisions, making them unreliable for safety-critical applications such as medical diagnosis (Ulmer et al., 2020) and self-driving cars (Filos et al., 2020). It has been pointed out that DNNs tend to assign high confidence scores to unknown inputs, which may result in incorrect predictions on the anomalous out-of-distribution (OOD) data (Nguyen et al., 2015). To tackle this problem, OOD detection has been actively investigated in the past few years (Hendrycks et al., 2022a; Yang et al.,

2022).

OOD detection aims at solving a K-class in-distribution (ID) classification task and a binary ID *vs.* OOD discrimination task at the same time. A commonly assumed practical setting is OOD examples are unavailable during training, which presents the major challenge for OOD detection. Most of recent research focuses on detecting visual OOD data, and only a few works (Hendrycks et al., 2020; Podolskiy et al., 2021; Zhou et al., 2021) study textual OOD detection. To our knowledge, current textual OOD detection methods typically apply a general OOD detection algorithm on representations yielded by Transformers (Vaswani et al., 2017) and are not tailored for textual data.

The main stream OOD detection methods commonly follow a post-hoc scheme (Hendrycks and Gimpel, 2017), which derives a OOD scoring function at the inference stage. The post-hoc scheme first discriminatively trains an ID K-class classifier by maximizing the conditional likelihood of $p(y|x)$, and then derives some statistics from the trained model to predictive OOD confidence scores. However, since the binary ID *vs.* OOD discrimination task is not considered in the training process, the learned representations by K-class training may be biased to the ID classes. While a few attempts (Hendrycks et al., 2019; Lee et al., 2018a) try to address this issue by introducing a surrogate OOD dataset during

the training stage, it is difficult to select proper surrogate datasets to represent the unknown large space of OOD data. Further, while the hierarchical contextual representation of pre-trained Transformers has been proven to be highly effective in various NLP tasks (Sun et al., 2019; Ma et al., 2019; Mohebbi et al., 2021; Devlin et al., 2019; Liu et al., 2019), its potential has yet to be well exploited for textual OOD detection.

To address the aforementioned problems, we propose a Transformer-based variational inference framework. Instead of only maximizing the conditional distribution $p(y|x)$ of ID data, we optimize the joint distribution $p(x, y)$, which is to maximize $p(y|x)$ and $p(x)$ simultaneously. The key idea is to model the distribution of the given ID data so as to leverage information that may not be useful for ID classification but crucial for outlier detection. To make the joint distribution $p(x, y)$ tractable, we resort to optimizing the evidence lower bound of $p(x, y)$ derived via amortized variational inference (AVI) (Kingma and Welling, 2014). In particular, based on the characteristics of textual data, we recast the posterior approximation distribution in AVI to condition on a dynamic combination of intermediate layer-wise hidden states of Transformers. The Transformer backbone acts like a shared encoder for both the ID classification head and the generator in AVI, as illustrated in Fig. 5.1.

We summarize the superiority of our method as follows.

- Our proposed variational inference framework for OOD detection (VI-OOD) provides a novel and principled perspective, which is orthogonal to previous textual OOD detection methods (Hendrycks et al., 2020; Podolskiy et al., 2021; Zhou et al., 2021).

- Our instantiation of VI-OOD exploits the rich contextual representation of pre-trained Transformers, which is tailored for textual OOD detection. It can learn better latent representations for text inputs, which can be readily applied to numerous existing post-hoc OOD detection algorithms and consistently improve their performance.

5.2 Problem Statement and Motivation

Out-of-distribution (OOD) detection aims to accurately separate all class-dependent in-distribution (ID) examples as well as out-of-distribution (or anomalous) examples. Given the input space $\mathcal{X} \times \mathcal{Y}$ and an ID class label set $\mathcal{Y}_{\text{ID}} = \{y_j\}_{j=1}^{K} \subset \mathcal{Y}$, an ID training set $\mathcal{D}_{\text{ID}} = \{(x_i, y_i)\}_{i=1}^{N}$ is sampled from the distribution $p(x, y)$ of ID data where $y_i \in \mathcal{Y}_{\text{ID}}$. With $\mathcal{D}_{\text{ID}}$, an ID classifier $f_{\text{ID}} : \mathcal{X} \to \mathcal{Y}_{\text{ID}}$ is trained. During test time, since there may be a distribution shift between the training and test data in practical application scenarios (Szegedy et al., 2014; Morningstar et al., 2021), the ID classifier f_{ID} may encounter OOD samples ($y_i \notin \mathcal{Y}_{\text{ID}}$).

Hence, an OOD confidence scoring function $f_{\text{OOD}} : \mathcal{X} \to \mathbb{R}$ is needed to perform ID *vs.* OOD binary classification. In this regard, OOD detection aims to solve both the K-class ID classification task and the binary outlier detection task. The ID classifier f_{ID} is commonly trained with a discriminative loss by maximizing the conditional log-likelihood of the training set:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \frac{1}{N} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{ID}}} \log p(y_i \mid x_i; f_{\text{ID}}, \theta), \tag{5.1}$$

where θ stands for all trainable parameters of f_{ID}.

The fundamental challenge of OOD detection is that at the training stage, real OOD examples are unavailable and thus cannot be effectively represented to provide necessary learning signals for the binary ID *vs.* OOD task. To address this issue, a few attempts have been made to introduce surrogate OOD datasets during training by using some datasets irrelevant to the ID data (Hendrycks et al., 2019; Lee et al., 2018a). However, it is difficult to select suitable "OOD" datasets to represent the huge space of real OOD data.

The majority of existing OOD detection methods (Hendrycks and Gimpel, 2017; Hendrycks et al., 2019; Lee et al., 2018b; Liu et al., 2020; Hendrycks et al., 2022a; Sun et al., 2021, 2022) follow a post-hoc paradigm and address the binary ID *vs.* OOD task in the inference stage.

These methods propose different OOD confidence scoring functions with the trained ID classifier f_{ID}. Specifically, the parameters of the trained f_{ID} are frozen, and some statistics of specific layers of f_{ID} (usually the penultimate layer or the softmax layer) are often used as OOD confidence scores.

While post-hoc methods have shown promise, it is pointed out that the performance of f_{ID} on ID data is not a good indicator of its performance on OOD data (Hendrycks et al., 2020; Lee et al., 2018a). Specifically, the discriminative training of f_{ID} is often conducted with $p(y|z)$, where z is the latent representation obtained by passing an input x to an DNN encoder. Maximizing the conditional log-likelihood $\log p(y|z)$ is essentially maximizing the mutual information between the latent variable Z and the label variable Y, i.e., $\mathcal{I}(Z, Y)$ (Boudiaf et al., 2020). Naturally, the learned representation Z will be biased towards the ID classification task. In fact, Kamoi and Kobayashi (2020) show that for the popular Mahalanobis distance based OOD detection method, the least important principal components of ID data is not useful for the ID classification task, but contain crucial information for the binary ID *vs.* OOD task, which may be discarded during the training of f_{ID} with $p(y|x)$.

To address this issue, we propose to learn better latent representation

Z for post-hoc methods by considering the distribution of ID data, i.e., maximizing the likelihoods $p(y|x)$ and $p(x)$ simultaneously[1], which is equivalent to modeling $p(x, y)$ – the joint distribution of ID data. To this end, we design a novel principled variational framework that will be elaborated in the next section.

5.3 Proposed Method

5.3.1 A Unified Variational Framework

Our goal is to directly maximize the likelihood of the joint distribution $p(x, y)$ rather than $p(y|x)$. We first define the learning problem from a probability perspective. We assume that a latent variable Z is a stochastic encoding of the input sequence X. The joint distribution $p(x, y, z)$ can be factored as:

$$\begin{aligned} p(x, y, z) &= p(y|z, x)p(x|z)p(z) \\ &= p(y|z)p(x|z)p(z), \end{aligned} \tag{5.2}$$

where we assume the Markov chain $X \leftrightarrow Z \leftrightarrow$ Y, i.e., $p(y|z, x) = p(y|z)$.

It is still intractable to compute $p(x, y)$ with Eq. (5.2). We employ amortized variational inference Kingma and Welling (2014) to solve the

[1]Note that $p(y|x) = \int_z p(y|z, x)p(z|x)$ and $p(x) = \int_z p(x|z)p(z)$.

problem. The log-likelihood of $p(x, y)$ can then be calculated by:

$$\begin{aligned}
\log p(x, y) &= \log \int_z p(x, y, z), \\
&= \log \int_z p(y|z)p(x|z)p(z)\frac{q(z|x)}{q(z|x)}, && (5.3) \\
&= \log \mathbb{E}_{z\sim q(z|x)}\left[\frac{p(y|z)p(x|z)p(z)}{q(z|x)}\right], && (5.4) \\
&\geqslant \mathbb{E}_{z\sim q(z|x)}\left[\log \frac{p(y|z)p(x|z)p(z)}{q(z|x)}\right], && (5.5)
\end{aligned}$$

where $q(z|x)$ in Eq. (5.3) is the amortized variational approximator of the true posterior $p(z|x)$. From Eq. (5.4) to Eq. (5.5), Jensen's inequality is applied. Eq. (5.5) can be rewritten as:

$$\begin{aligned}
\mathcal{L}_{\text{ELBO}} =& \mathbb{E}_z\left[\log p(y|z)\right] + \mathbb{E}_z\left[\log p(x|z)\right] - \\
& D_{\text{KL}}(q(z|x)||p(z)). && (5.6)
\end{aligned}$$

At this point, the evidence lower bound of $p(x, y)$ has been derived, denoted by $\mathcal{L}_{\text{ELBO}}$, where the first term corresponds to the ID supervised training target, and the second and third terms correspond to the unsupervised learning target for amortized variational Bayesian autoencoder.

5.3.2 Instantiation

Next, we instantiate the proposed variational framework for textual OOD detection, which consists of a discriminator $p(y|z)$, a decoder (or

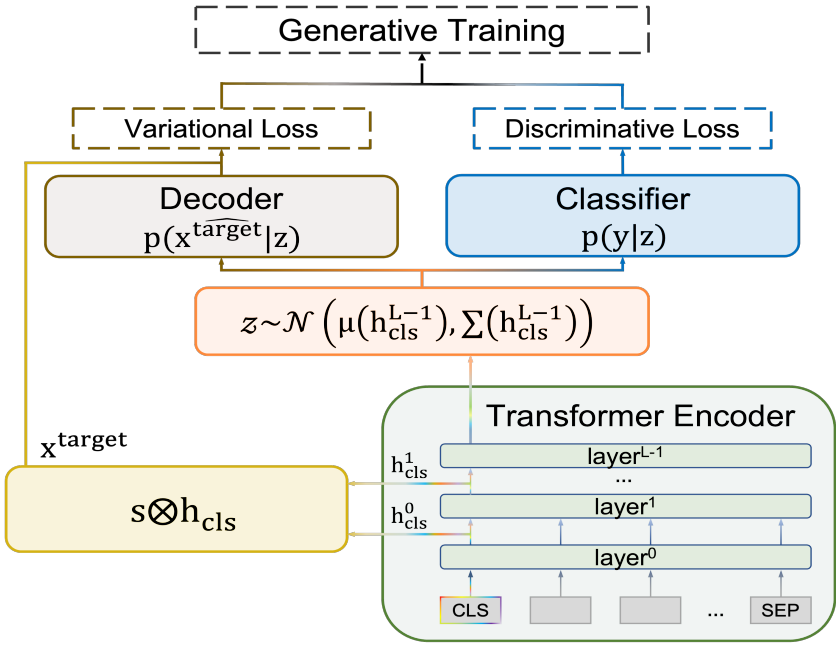

Figure 5.1: The architecture of our proposed framework. Our method employs an encoder-based Transformer model as the backbone textual encoder. Hidden states of the [CLS] token are chosen to be textual representations. z is a latent variable conditioned on the textual representations. The in-distribution (ID) classification head $p(y|z)$ and decoder $p(x^{\hat{\text{target}}}|z)$ both take z as the input. s is the hidden states combination factor and the merge representation x^{target} works as the target of the decoder.

reconstructor) $p(x|z)$, and a posterior approximator (or encoder) $q(z|x)$. For image data, the decoder target is commonly defined as the original input x, which is natural since it is most informative. However, for textual data, the input token sequences are merely the embedding vectors from a chosen dictionary, while the intermediate hidden states of Transformers may contain ample contextual semantics. Hence, it is nontrivial to define the reconstruction target of $p(x|z)$ and the latent variable z for textual data.

Textual representation Contextual representations of a sentence or a paragraph are usually extracted by encoder-based Transformers. In this paper, we use BERT family models Devlin et al. (2019). Let the original x be a sequence of tokens, i.e., $[x_0, \cdots, x_{s-1}]$ with length s. BERT adds the special token [CLS] at the beginning of the input sequence for classification tasks, i.e., $[\text{CLS}, x_0, \cdots, x_{s-1}]$. Unless otherwise specified, we use the hidden states of the [CLS] token as the text representation. The input sequence is passed through each layer of BERT, outputting a series of intermediate hidden states corresponding to [CLS] position, denoted as $h_{\text{CLS}} = [h^0_{\text{CLS}}, \cdots, h^{L-1}_{\text{CLS}}]$, where L is the total number of layers.

Training As shown in Fig. 5.1, we instantiate the encoder $q(z|x)$ and the prior $p(z)$ as diagonal Gaussian distributions, i.e., $\mathcal{N}(z|\mu, \Sigma)$ and $\mathcal{N}(0, I)$ respectively, where μ and Σ are obtained by mapping the last hidden state h^{L-1}_{CLS} with a single-layer MLP. Besides, we introduce a weight vector $\mathbf{s} \in \mathbb{R}^L$ to dynamically integrate the intermediate hidden states of the Transformer. Then, we derive the reconstruction target $x^{\text{target}} = h_{\text{CLS}} \otimes \mathbf{s}$ ($\otimes$ denotes element-wise multiplication), which contains rich contextual semantic information. Referring to Fig. 5.1, the reconstructor $p(x|z)$ takes a sample z from $\mathcal{N}(z|\mu, \Sigma)$ as input and out-

puts a reconstructed version of x^{target} to maximize $p(x^{\text{target}}|z)$. The ID classifier f_{ID} is a single-layer MLP which takes the latent representation z as input.

Inference At the inference stage, we only need the trained posterior approximator (encoder) $q(z|x)$ and the ID classifier f_{ID}. Note that both the ID classification task and the binary outlier detection task are performed w.r.t. the latent variable z. For each x, we only sample one z during training and inference respectively.

5.4 Experiments

In this section, we present a comprehensive analysis for textual out-of-distribution (OOD) detection with various transformers and pervasive OOD detection methods. Besides, we demonstrate the effectiveness of our proposed OOD detection method on challenging natural language understanding benchmarks. We start this section by describing our evaluation methodology and then present our experimental results in the following.

5.4.1 Evaluation Methodology

Datasets OOD detection in the natural language processing(NLP) domain is generally under-explored and only discuss in limited scenar-

ios such as out-of-scope intent detection in dialogue machines (Zhan et al., 2021; Zhang et al., 2021; Yan et al., 2020). As such, evaluating OOD performance in the NLP domain dose not have a consensus. To scale the evaluation process as general as possible, we follow the evaluation in (Hendrycks et al., 2020; Zhou et al., 2021) to present our main analysis. Hendrycks et al. (2020) firstly propose to use the sentiment analysis benchmark SST-2 as the in-distribution dataset and select five other datasets as out-distribution evaluation sets, which includes 20 Newsgroups, WMT16 and Multi30K, RTE and SNLI. Zhou et al. (2021) further extend this benchmark by adding more natural language understanding tasks includes topic classification, question classification.

In-distribution Tasks We use the bellowing four benchmark datasets as in-distribution (ID) tasks. When setting each of them as *in-distribution*, other ones are recognized as *out-distribution*.

- 20 Newsgroups (20NG) (Lang, 1995) is a commonly used benchmark for the topic (or newsgroup) classification. In this dataset, these are 20 labeled topic classes and $15,056$, $1,876$ and $1,896$ examples for the train, validation and test respectively.

- IMDB (Maas et al., 2011) contains $25,000$ movie reviews from IMDB which is collected for the task of binary sentiment analysis.

10% examples are randomly selected as the validation set.

- SST-2 (Socher et al., 2013) is also a binary sentiment analysis task derived from the Stanford Sentiment Treebank. It consists of $67,349$, 872 and $1,821$ examples for training, validation and test sets respectively. Note that since both IMDB and SST-2 are sentiment analysis datasets, they are not considered as OOD counterparts in our experiments.

- TREC-10 (Li and Roth, 2002) is a dataset for question classification, which is a preliminary task for question answering. We also use the 6-class version as Zhou et al. (2021) did. The splitting is $4,907$ for training, 545 for validation and 545 for test.

Besides the above ID four tasks, we also use another four unrelated datasests as OOD test sets (not for training) for all of the four ID tasks. We refer them as the out-distribution datasets: the English source side of English-German WMT16 (Bojar et al., 2016) and English-German Multi30K (Elliott et al., 2016), and concatenations of the premise and hypothesis of RTE (Dagan et al., 2006) and MNL (Williams et al., 2018). WMT16 and Multi30K are for machine translation while RTE and MNLI are for natural language inference. We use the respective test sets of each out-distribution dataset to measure OOD performance.

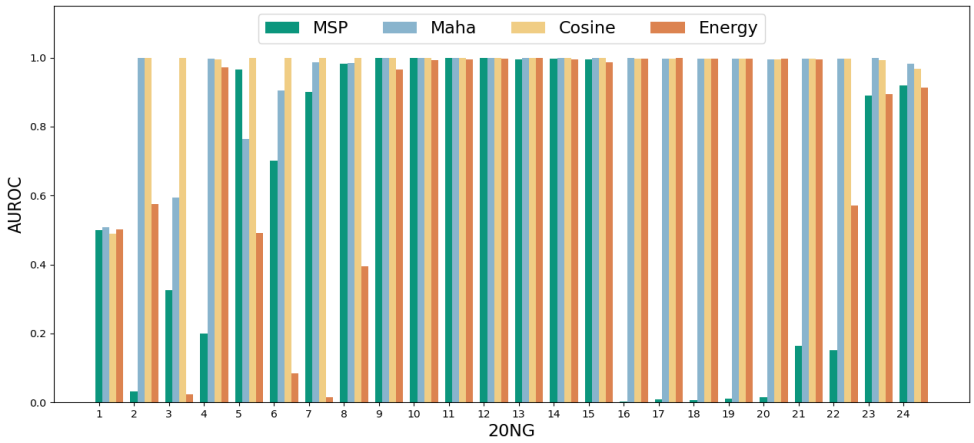

Figure 5.2: A study on the OOD performance of the intermediate hidden sates. AUROC results across 24 layers of RoBERTa$_{\text{LARGE}}$ are reported (higher represents better). The model is fine-tuned on SST-2 and evaluates OOD performance on 20NG. Intermediate layers 9 to 15 could bring more benefits to four popular OOD detectors (marked by green, blue, light yellow and orange) than the last hidden states (layer 24). Hence, exploiting the intermediate hidden states in a more efficient way will provide significant improvement for textual OOD detection.

Baselines To demonstrate the challenges and characteristics of OOD detection with Transformers, we present baseline results for the following four state-of-the-art OOD detection methods:

- **Maximum Softmax Probability (MSP)** (Hendrycks and Gimpel, 2017): The MSP confidence score leverages the maximum softmax probability outputted by the softmax function for out-of-domain detection. As correct samples tend to have higher probability scores, samples below a threshold are more likely to be outliers. Specifically, the Confidence score is $\mathcal{C}(x) = \max_y p(y|x)$.

- **Mahalanobis Distance (Maha)** (Lee et al., 2018b): The Mahalanobis Distance (MD) method fits K-class conditional Gaussian distributions $\{\mathcal{N}(\mu_i, \Sigma)\}_{i=1}^{K}$ for the K in-distribution classes upon the output of the penultimate layer in the model. The Mahalanobis Distance and the MD confidence score are computed by:

$$\begin{aligned} \text{MD}_k(z) &= (z - \mu_k)^T \Sigma^{-1} (z - \mu_k), \\ \mathcal{C}(x) &= -\min_k \{\text{MD}_k(z)\}. \end{aligned} \tag{5.7}$$

- **Energy score (Energy)** (Liu et al., 2020): The energy score confidence score is inspired by the energy-based models LeCun et al. (2006). It defines an energy of an input (x, y) as $E(x, y) = w_y^T \cdot z$, where w_y is the weight of the softmax layer for the y^{th} in-distribution class. The energy score confidence score is defined as:

$$\mathcal{C}(x) = \log \sum_{i}^{K} e^{w_i^T \cdot z}. \tag{5.8}$$

- **Cosine distance (Cosine)** (Zhou et al., 2021): The cosine distance OOD confidence sore defines as the maximum cosine similarity of a test input representation with representations in the validation set, i.e., $\mathcal{C}(x) = -\max_{i=1}^{V} \cos(z, z_i^{val})$.

Methods	SST-2			IMDB		
	AUROC ↑	FAR@95 ↓	AUPR ↑	AUROC ↑	FAR@95 ↓	AUPR ↑
MSP	89.85	66.20	86.40	94.30	41.90	98.80
MSP$_{VI}$	92.85 (+3.00)	51.58 (+14.62)	89.72 (+3.32)	95.95 (+1.65)	28.03 (+13.87)	99.12 (+0.32)
Maha	97.98	11.50	97.30	99.67	0.70	99.95
Maha$_{VI}$	**99.33** (+1.35)	**3.62** (+7.88)	**98.52** (+1.22)	**99.90** (+0.23)	**0.21** (+0.49)	**99.97** (+0.02)
Cosine	95.65	22.65	94.68	99.50	1.53	99.88
Cosine$_{VI}$	98.87 (+3.22)	6.62 (+16.03)	98.06 (+3.38)	99.57 (+0.07)	1.43 (+0.10)	99.88 (0.00)
Energy	89.80	67.00	86.53	93.30	56.70	98.63
Energy$_{VI}$	92.79 (+2.99)	51.25 (+15.75)	89.26 (+2.73)	96.05 (+2.75)	27.97 (+28.73)	99.12 (+0.49)

Methods	TREC-10			20NG		
	AUROC ↑	FAR@95 ↓	AUPR ↑	AUROC ↑	FAR@95 ↓	AUPR ↑
MSP	97.94	8.43	89.26	93.89	30.49	87.39
MSP$_{VI}$	98.91 (+0.97)	2.77 (+5.66)	90.39 (+1.13)	93.29 (-0.60)	25.61 (+4.88)	80.09 (-7.3)
Maha	98.99	4.87	95.11	98.39	7.77	95.91
Maha$_{VI}$	**99.46** (+0.47)	**0.79** (+4.08)	**97.67** (+2.56)	**99.80** (+1.41)	**0.61** (+7.16)	**98.93** (+3.02)
Cosine	98.89	3.96	94.54	97.73	10.84	88.71
Cosine$_{VI}$	99.36 (+0.47)	1.19 (+2.77)	96.09 (+1.55)	99.39 (+1.66)	2.92 (+7.92)	97.19 (+8.48)
Energy	97.19	10.07	82.16	95.76	17.93	88.71
Energy$_{VI}$	99.21 (+2.02)	2.84 (+7.23)	90.84 (+8.68)	94.34 (-1.42)	17.04 (+0.89)	79.67 (-9.04)

Average	AUROC ↑	FAR@95 ↓	AUPR ↑
avg. (MSP / Maha / Cosine / Energy)	94.00 / 98.78 / 97.94 / 94.01	36.76 / 6.21 / 9.75 / 37.93	**90.46** / 97.07 / 94.45 / 89.01
avg.$_{VI}$ (MSP / Maha / Cosine / Energy)	**95.25** / **99.62** / **99.30** / **95.60**	**27.00** / **1.31** / **3.04** / **24.78**	89.83 / **98.77** / **97.81** / **89.72**

Table 5.1: Main results of our proposed variational inference (VI) framework. MSP, Maha, Energy and Cosine are baseline methods trained with the discriminative loss while each corresponding method with the *VI* subscript denotes the model trained with our VI framework. The best result is marked in bold. Models are fine-tuned on the training set of each in-distribution (ID) datasets, i.e., SST-2, IMDB, TREC-10 and 20NG. At the bottom row, averaged results across four ID datasets are included. Results for each metrics are averaged across 8 OOD test datasets. All results are percentages.

Metrics We employ three commonly used metrics for OOD detection and introduce them as follows:

- **AUROC**: Area Under the Receiver Operating Characteristic curve (AUROC) reveals the relationship between True Positive Rate (TPR) (i.e., Recall) and False Positive Rate (FPR). It represents the probability of assigning a higher score to a positive example than a negative examples. The pioneering work Hendrycks and Gimpel

(2017) firstly proposed to use this metric for OOD detection. A higher AUROC score indicates a better classifier, and An AUROC score of 50% means random guessing.

- **FAR@95**: False Alarm Rate at 95% Recall (FAR@95) is the probability that a negative example is misclassified as positive when Recall or TPR is 95%. In this paper, we take the OOD class as negative.

- **AUPR**: Area Under the Precision-Recall curve (AUPR) is another commonly used metric based on the Precision-Recall Curve. It is a better indicator in the case of imbalanced in- and out-rate Manning and Schutze (1999). A perfect classifier has an AUPR of 100%.

Experimental Setup We employ the RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019) model from the HuggingFace library (Wolf et al., 2019) as our main backbone to conduct experiments. We use the optimizer AdamW (Loshchilov and Hutter, 2019) with a linear-scheduled learning rate $1e-5$ to fine-tune the model for 20 epochs. For the variational terms in Eq. 5.6, we apply a linear annealing strategy which is a common practice in variational methods (Fu et al., 2019). All reported results are obtained in 5 runs with different random seeds.

5.4.2 A Closer Look at OOD Detection with Transformers

In Fig. 5.2, we conduct in-depth experiments to investigate the impacts of the intermediate hidden states of $\text{RoBERTa}_{\text{LARGE}}$ to OOD detection. Following Hendrycks et al. (2020), we take the model trained on SST-2 as a case study. Here, the model is trained solely with the discriminative loss. We perform OOD detection on each of the 24 hidden states of the trained model and summarize the AUROC results in Fig. 5.2. As shown in Fig. 5.2, we report the aforementioned four baselines for each hidden layer. As the layer number increases from 1 to 24, the corresponding hidden layer is closer to the head of the model, i.e., layer 24 outputs the last hidden states.

Intermediate hidden states could help OOD detection. As shown in Fig. 5.2, it is obvious that for all the four OOD detection score functions, the intermediate hidden states could achieve better OOD performance than the last hidden states. In particular, we can observe that the patterns of the OOD performance for all the OOD test datasets are very similar. All the four OOD score functions achieve best OOD performance around the middle layers (layer 9 to 13) and the performance is consistent among these adjacent layers. In addition, we surprisingly find that the most

straightforward OOD score function – Cosine – starts to work at layer 2, which is faster than the other three functions. These observations validate our key assumption that information unrelated to ID classification could help with OOD detection.

Gaps between different OOD detection score functions could be filled. Performance on layer 24 or 23 which are the top layers of the model is very similar with respect to the four baselines. Specifically, Maha OOD score function achieves the best performance across all the OOD datasets. The Cosine OOD score function could achieve close results to Maha but is still a bit inferior. MSP and Energy perform similarly but far behind Maha and Cosine. However, when looking at intermediate layers, the performance gaps among different OOD score functions became tiny. For example, MSP could achieve the near-best results around layer 13 for the two OOD datasets. These observations suggest that properly exploiting the power of hidden states of Transformers could reduce the difficulty of the OOD detection problem.

5.4.3 Main Results

We demonstrate the versatility of our proposed VI-OOD detection framework by considering four in-distribution (ID) datasets in addition to the other four out-distribution datasets, i.e., the model trained on each ID

dataset will be evaluated on seven OOD test sets (except SST-2 and IMDB). Averaged Results are summarized in Table 5.1 and detailed results are presented in Table 5.3 and Table 5.4 in the appendix.

VI-OOD benefits a diverse collection of tasks and OOD score functions. According to Table 5.1, A salient observation is that for all the compared OOD score functions, our proposed approach can consistently achieve better performance. For example, for the best performing baseline – the Maha method, our method reduces the average FAR@95 from 6.21% to 1.31%, which leads to a 78.9% relative increase. For the second best baseline–Cosine score function, the improvement of our method is also significant, i.e., reducing average FAR@95 from 9.75% to 3.04%. Besides the improvements on FAR@95, performance gains on AUROC are also significant. For example, the average AUROC sore of the Cosine method increases from 97.94% to 99.3%. Considering that out method requires no real OOD examples involved, these results are very encouraging.

Looking into each of the four ID datasets, it can be seen that detecting OOD test examples on the model trained on TREC-10 is much easier than other datasets, i.e., all OOD score functions achieve the AUROC score above 97%. Improvements upon these competitive results can be

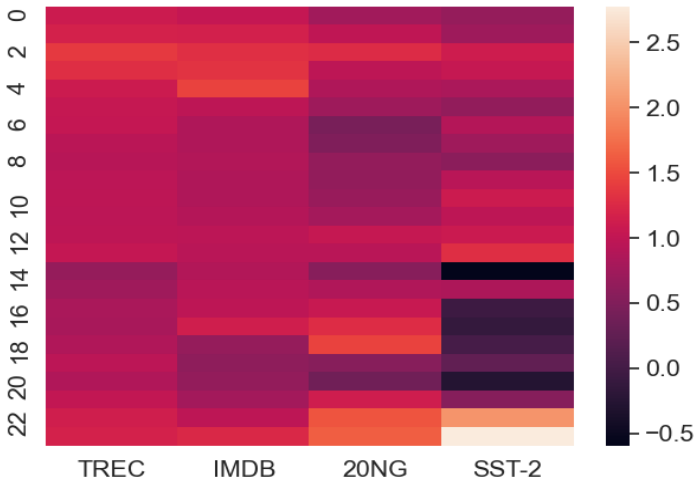

Figure 5.3: Heatmap of the hidden state combination factor s. The horizontal axis stands for four ID task and the vertical axis represents layer number.

challenging. Nevertheless, our method also achieves better performance for all four score functions. In particular, The AUROC score of Energy is improved from 97.19% to 99.21% and the corresponding FAR@95 score is reduced from 10.07% to 2.84%. For the Maha method on TREC-10, our method reduces the FAR@95 score to 0.79% which is near perfect.

5.4.4 ID classification Performance

In this subsection, we investigate the ID classification performance. Besides the binary ID VS OOD task, OOD detection also concerns the ID classification task. We summarize the test accuracy of the corresponding ID test sets for the four ID datasets in Table 5.2. It can be seen that for 20NG, SST-2 and TREC-10, IMDB, ID test performances are very

Test Accuracy	SST-2	IMDB	TREC-10	20NG
$p(y\|x)$	96.21	95.33	97.8	93.99
$p(x, y)$	96.38	94.54	97.0	93.35

Table 5.2: Performance comparison of the ID K-class classifier for different training objectives. $p(y|x)$ is the commonly used discriminative objective and $p(x, y)$ is our proposed objective.

similar and all the gaps are lower than 1%. Therefore, models trained with our proposed $p(x, y)$ target do not bring significant detrimental impacts to ID classification. However, although we consider these gaps can be ignored in practical applications, it also indicates that our method can be further improved in further works.

5.4.5 The Combination Factor s

At last, we analyze the proposed combination factor s. We plot the heatmap of the learned s for in-distribution (ID) tasks in Fig. 5.3. It can be seen that for different ID tasks, the hidden state combination patterns are very different. It verifies the flexibility of our proposed framework in utilizing the powerful hidden states of pre-trained models.

5.4.6 Detailed Experimental Results

In the following, we summarize detailed experimental results for baselines in Table 5.3 and for our proposed framework in Table 5.4.

AUROC	SST-2				IMDB				TREC-10				20NG			
	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy
SST-2	-	-	-	-	-	-	-	-	96.1	97.2	97.5	92	97.9	99.6	99.4	99.1
IMDB	-	-	-	-	-	-	-	-	99	99.4	99.2	98.8	96.9	99.1	98.8	98.1
TREC-10	92.7	95.1	88.1	93.1	95.7	99.7	99.8	94.2	-	-	-	-	91.9	98.8	97.8	94.6
20NG	92.6	96.6	93.3	92.6	96.2	99.7	99.5	96.6	98.5	99.4	99.4	99.1	-	-	-	-
Multi30k	90.5	99.2	97.9	91.2	96	99.7	99.6	94.7	98.9	99.6	99.5	98.9	95.8	98.9	98.5	97.4
RTE	89.9	99.8	99.7	88.7	92.9	99.7	99.6	91.6	97.6	99.3	99	97	88.7	96.2	95	90.8
WMT16	86.7	98.7	97.5	86.2	92.6	99.7	99.4	91.2	98.2	99.3	99.1	97.7	91.7	97.7	96.8	94.1
MNLI	86.7	98.5	97.4	87	92.4	99.5	99.1	91.5	97.3	98.7	98.5	96.8	94.3	98.4	97.8	96.2
AVG.	89.85	97.98	95.65	89.8	94.3	99.67	99.5	93.3	97.94	98.99	98.89	97.19	93.89	98.39	97.73	95.76
FAR@95	SST-2				IMDB				TREC-10				20NG			
	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy
SST-2	-	-	-	-	-	-	-	-	19.2	18.4	10.3	28.6	11.5	1.3	2.5	4.8
IMDB	-	-	-	-	-	-	-	-	2.5	1.4	1.6	3	16.5	3.4	4.8	8
TREC-10	54.8	26.2	59.4	52.4	34.6	0	0	58.8	-	-	-	-	47.2	5.8	12.2	26.6
20NG	61.2	23.7	44.9	59.3	25.2	0.89	1.2	29.3	5.2	2.2	2.6	4	-	-	-	-
Multi30k	63.1	4.3	8.7	56.8	26.8	0.1	0.8	48.3	2.9	1.2	1.2	3.9	19.1	5.4	6.6	9.6
RTE	72.4	0.4	0.67	82.8	57.8	0.8	1.3	71.9	12.1	2.8	3.6	12.8	46.7	17.2	20.9	31.4
WMT16	74.7	6.9	10.4	79.3	53.8	1.1	2	68.3	6.8	2.3	2.8	7.3	41.6	13.1	18	27.3
MNLI	71	7.5	11.8	71.4	53.2	1.3	3.8	63.6	10.3	5.8	5.6	10.9	30.8	8.2	10.9	17.8
AVG.	66.2	11.5	22.65	67	41.9	0.7	1.527	56.7	8.43	4.87	3.96	10.07	30.49	7.77	10.84	17.93
AUPR	SST-2				IMDB				TREC-10				20NG			
	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy
SST-2	-	-	-	-	-	-	-	-	90.6	94.5	95.3	75.7	98.2	99.7	99.5	99.1
IMDB	-	-	-	-	-	-	-	-	91.8	95.1	94.3	88.6	76.5	90.8	89.6	77.7
TREC-10	98.1	98.6	96.6	98.2	100	100	100	99.9	-	-	-	-	97.6	99.7	99.4	98.3
20NG	94.9	97.3	94.7	94.7	99.7	100	100	99.7	96.2	98.6	98.5	97.1	-	-	-	-
Multi30k	89.7	99	97.1	89.8	99.6	100	100	99.5	96.9	98.8	98.4	94.3	94.5	98.6	98.1	96.1
RTE	90.2	99.7	99.6	89.6	99.2	100	99.9	99	93.5	97.9	96.8	86.3	81.4	94.4	92.6	82.3
WMT16	85	97.9	95.7	84.9	99.1	100	100	98.9	93.3	97.7	97	86.7	89.3	96.9	95.8	91.2
MNLI	60.5	91.3	84.4	62	95.2	99.7	99.4	94.8	62.5	83.2	81.5	46.4	74.2	91.3	88.7	76.3
AVG.	86.4	97.3	94.68	86.53	98.8	99.95	99.88	98.63	89.26	95.11	94.54	82.16	87.39	95.91	94.81	88.71

Table 5.3: The OOD performance of baseline models trained by the discriminative loss. Models are fine-tuned on the training set of each in-distribution datasets, i.e., SST-2, IMDB, TREC-10 and 20NG. The OOD metrics are calculated by treating each dataset in the first column as the OOD dataset.

5.5 Chapter Review

In this chapter, we have investigated OOD detection with Transformers for NLP classification tasks. We propose a variational Bayesian framework, namely VI-OOD, to optimize the joint distribution $p(x, y)$ for model training. We provide both experimental evidence and theoretical insights for our proposed approach. Comprehensive experiments on large-scale NLP tasks have validated the effectiveness and superiority of our novel OOD framework. In addition, we provide analysis on the hidden states of Transformers, which may shed new light on textual

AUROC	SST-2				IMDB				TREC-10				20NG			
	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy
SST-2	-	-	-	-	-	-	-	-	99.1	98.5	98.7	99.3	97.9	100	100	98.7
IMDB	-	-	-	-	-	-	-	-	99.6	99.6	99.7	99.8	96.6	99	98.8	97.4
TREC-10	95.43	98.9	97.94	94.93	98.1	100	100	97.9	-	-	-	-	93.8	100	100	95.5
20NG	95.74	99.88	99.4	95.53	96.3	99.9	99.5	96.7	99.2	99.4	99.2	99.5	-	-	-	-
Multi30k	94	99.3	98.4	94.37	97.1	100	99.8	97.3	99.6	99.8	99.8	99.7	96.7	100	100	97.6
RTE	93.06	100	99.9	93.27	95	99.9	99.5	95.1	99.3	99.8	99.7	99.5	85.2	99.7	98.6	85.7
WMT16	89.56	99.05	98.51	89.4	94.9	99.8	99.5	95	98.4	99.6	99.4	98.5	90.1	100	99.2	91.5
MNLI	89.31	98.83	99.08	89.23	94.3	99.8	99.1	94.3	97.2	99.5	99	98.2	92.7	99.9	99.1	94
AVG.	92.85	99.33	98.87	92.79	95.95	99.9	99.57	96.05	98.91	99.46	99.36	99.21	93.29	99.8	99.39	94.34
FAR@95	SST-2				IMDB				TREC-10				20NG			
	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy
SST-2	-	-	-	-	-	-	-	-	2.3	2.7	1.9	2.6	9.1	0	0.16	4.6
IMDB	-	-	-	-	-	-	-	-	0.5	0.1	0.3	0.6	14.8	2.6	4.7	7.9
TREC-10	39	6.6	13.2	42.6	7.4	0	0	9.2	-	-	-	-	24.4	0	0.4	13.2
20NG	37.39	0.11	3.1	39.08	25	0.42	1.05	23.8	1.6	0.05	1.4	1.7	-	-	-	-
Multi30k	44.35	3.32	10.7	40.88	13.2	0	0.63	12.6	1.1	0.07	0.1	1.2	12.9	0	0.07	6.5
RTE	57.87	0	0	55.53	41.2	0.26	1	40.6	1.7	0.1	0.3	1.9	48.4	1.2	6.8	37.5
WMT16	65.66	5.26	7.84	64.99	39.2	0.2	2.2	39.1	4.4	0.9	1.6	5.1	38.7	0.17	3.8	28.6
MNLI	65.22	6.42	4.88	64.4	42.2	0.39	3.7	42.5	7.8	1.6	2.7	6.8	31	0.32	4.5	21
AVG.	51.58	3.62	6.62	51.25	28.03	0.21	1.43	27.97	2.77	0.79	1.19	2.84	25.61	0.61	2.92	17.04
AUPR	SST-2				IMDB				TREC-10				20NG			
	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy	MSP	Maha	Cosine	Energy
SST-2	-	-	-	-	-	-	-	-	96.8	97.7	97.6	97.1	97.2	100	99.9	97.5
IMDB	-	-	-	-	-	-	-	-	93.4	98.1	97.1	93.2	64.8	93.7	90.7	63.8
TREC-10	98.84	99.71	99.43	98.74	100	100	100	100	-	-	-	-	97.7	100	100	98.1
20NG	96.92	99.89	99.44	96.78	99.7	100	100	99.7	98.3	99.1	98.5	98.5	-	-	-	-
Multi30k	93.9	99.13	98.02	94.1	99.7	100	100	99.7	98.6	99.4	99.4	98.2	94	100	100	94.5
RTE	92.85	100	99.9	92.82	99.4	100	99.9	99.4	96.4	99.5	99.1	96.5	71.5	99.6	97.9	70
WMT16	87.81	98.61	98.12	87.1	99.4	100	100	99.4	88	98.7	97.9	86.5	81.9	100	98.6	82.7
MNLI	68.01	93.78	93.46	65.99	96.5	99.8	99.4	96.5	61.2	91.2	83	65.9	53.5	99.2	93.2	51.1
AVG.	89.72	98.52	98.06	89.26	99.12	99.97	99.88	99.12	90.39	97.67	96.09	90.84	80.09	98.93	97.19	79.67

Table 5.4: The OOD performance of our proposed variational framework trained by generative loss. Models are fine-tuned on the training set of each in-distribution datasets, i.e., SST-2, IMDB, TREC-10 and 20NG. The OOD metrics are calculated by treating each dataset in the first column as the OOD dataset.

OOD detection.

This research focuses on improving AI safety and model robustness. Therefore, our work can benefit a variety of AI applications, and there is no direct risk of abuse. Besides, our proposed method only uses open-sourced benchmarks as training data. It does not introduce additional datasets for training the OOD detector, thus having no ethical issues in collecting datasets. However, since this study takes the pre-trained language model RoBERTa$_{\text{LARGE}}$ as the backbone encoder, the obtained results may be affected by the various biases of the pre-trained language

model.

Chapter 6

Conclusion and Future Works

Conclusion. In this thesis, we have developed novel learning frameworks and algorithms for textual out-of-distribution (OOD) detection by leveraging the pre-trained Transformers' representation capabilities. Our discoveries have introduced new and promising prospects for textual OOD detection. Importantly, all of our proposed methods aim to enhance OOD detection performance in a self-supervised or unsupervised manner, thereby avoiding the need for additional datasets. Consequently, our approaches are more applicable to real-world scenarios.

We first identify the major weakness of previous methods is that the traditional learning paradigm requires manually selecting a proper

threshold for OOD discrimination. Our self-supervised learning framework, as described in Chapter 3, demonstrates the feasibility of directly training a self-supervised (K+1)-way classifier for end-to-end textual OOD detection without requiring human intervention. This approach showcases that OOD data can be effectively represented in Transformer's representation space by linear interpolation between in-distribution (ID) samples.

Further, in Chapter 4, we investigate low-resource scenarios and find that the representation space of Transformers exhibits smoothness with regards to semantics. This means that representations in the proximity of an actual in-distribution (ID) example are highly similar and interconnected in terms of their semantics. Therefore, we demonstrate that a lightweight denoising autoencoder (DAE) can be trained in the representation space of Transformers to capture the local structure of ID examples. The DAE can effectively produce ID representatives that assist the (K+1)-way classifier in establishing appropriate decision boundaries for ID classification.

Finally, our thesis delves deeper into the training objective of previous out-of-distribution (OOD) detection methods and identified that ID discriminative training is biased towards the ID task, ignoring the ID *vs.* OOD task. Our in-depth analysis of the OOD characteristics of interme-

diate layers in Transformers reveal that these layers can be utilized to improve OOD detection. In order to design an unbiased objective and better leverage the hierarchical representations in Transformers, we developed a principled variational learning framework specifically tailored for textual OOD detection. Our key idea was to maximize the likelihood of the joint distribution $p(x, y)$. To achieve this, we reformulated the architecture of a classic amortized variational autoencoder, taking the entire Transformer as the encoder and using a dynamic combination of the hierarchical representations as the reconstruction target. Through extensive experiments, we demonstrated that our approach can learn better representations for textual OOD detection and consistently improve popular post-hoc OOD detectors.

The significance of this thesis is two-fold:

- The thesis undertakes a comprehensive and methodical investigation of textual OOD detection, an area that has received relatively little attention despite its significant importance to machine learning safety.

- The thesis leverages the contextualized representations of Transformers to develop effective self-supervised and unsupervised learning frameworks for textual OOD detection. These frameworks not only contribute to improving textual OOD detection but

also offer novel insights into general OOD detection.

Future Works. The subsequent matters are left for future investigation.

1. Exploring textual augmentation. As discussed in Chapter 2, recent developments in visual OOD detection have demonstrated that data augmentation methods, such as mixing images with fractal images, can significantly enhance the reliability of deep neural networks. Fractal images, with their inherent complexity, can compel the model to learn more robust features rather than shortcut features, which can benefit a range of machine learning safety goals, including OOD detection. While this approach is promising, defining the notion of "complexity" for textual sequences is ambiguous. In other words, it is difficult to find sentences that are evidently complex since textual data is subjective, and it is nearly impossible to define and annotate complex texts manually. However, in the future, a promising solution could be to utilize large-scale pre-trained language models (PLMs) like ChatGPT to generate complex texts that can be combined with original texts. These PLMs, specifically designed for natural language generation purposes, have shown to be highly effective in interpreting, generating language, and open-domain dialogue.

2. Applications to general OOD detection. The potential of Transformer-based models in computer vision has been demonstrated in recent advancements (Liu et al., 2021; Khan et al., 2022; Liang et al., 2021; Bao et al., 2021). These models have been successfully applied to various vision tasks, achieving new state-of-the-art performance by reformulating the tasks to fit Transformer-based models. Our proposed self-supervised learning and probabilistic representation learning frameworks can potentially improve the performance of visual OOD detection using vision Transformers. Furthermore, the proposed lightweight ID data augmentation method can also be applied to other types of data, such as audio and images, to generate synthetic data and improve the performance of OOD detection models. Additionally, with the growing interest in vision-language transformers, our approaches can also be evaluated on multi-modal OOD detection tasks.

Bibliography

Charu C Aggarwal and Philip S Yu. 2001. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 37–46.

Abhinav Arora, Akshat Shrivastava, Mrinal Mohit, Lorena Sainz-Maza Lecanda, and Ahmed Aly. 2020. Cross-lingual transfer learning for intent detection of covid-19 utterances.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. 2020. A unifying mutual information view of metric learning: cross-entropy vs. pairwise

losses. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 548–564. Springer.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Hyunsun Choi, Eric Jang, and Alexander A Alemi. 2018. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning*

Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment: First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers, pages 177–190. Springer.

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

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Angelos Filos, Panagiotis Tigas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. 2020. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 240–250. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. 2022a. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 8759–8773. PMLR.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. 2021b. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting

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

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. 2019. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. 2022b. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16762–16771. IEEE.

Rui Huang, Andrew Geng, and Yixuan Li. 2021. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34:677–689.

Ryo Kamoi and Kei Kobayashi. 2020. Why is the mahalanobis distance effective for anomaly detection? *CoRR*, abs/2003.00402.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41.

Sopan Khosla and Rashmi Gangadharaiah. 2022. Evaluating the practical utility of confidence-score based techniques for unsupervised open-world classification. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 18–23.

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

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann.

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

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018a. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018b. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. 6th International Conference on Learning Representations, ICLR 2018 ; Conference date: 30-04-2018 Through 03-05-2018.

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

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the*

IEEE/CVF international conference on computer vision, pages 10012–10022.

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

Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2019. Universal text representation from BERT: an empirical study. *CoRR*, abs/1910.07973.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

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

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Markos Markou and Sameer Singh. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497.

Alexander Meinke and Matthias Hein. 2020. Towards neural networks that provably know when they don't know. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 983–992. ACM.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Hosein Mohebbi, Ali Modarressi, and Mohammad Taher Pilehvar. 2021. Exploring the role of BERT token representations to explain sentence probing results. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 792–806. Association for Computational Linguistics.

Warren R. Morningstar, Cusuh Ham, Andrew G. Gallagher, Balaji Lakshminarayanan, Alexander A. Alemi, and Joshua V. Dillon. 2021. Density of states estimation for out of distribution detection. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 3232–3240. PMLR.

Norman Mu and Justin Gilmer. 2019. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*.

Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. 2019. Do deep generative models know

what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 427–436. IEEE Computer Society.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A*

meeting of SIGDAT, a Special Interest Group of the ACL, pages 1532–1543. ACL.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

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

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2008. *Dataset shift in machine learning*. Mit Press.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020.

Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. 2021. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*.

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.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE.

Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. 2021. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795.

Seonghan Ryu, Seokhwan Kim, Junhwi Choi, Hwanjo Yu, and Gary Geunbae Lee. 2017. Neural sentence embedding using only in-domain

sentences for out-of-domain sentence detection in dialog systems. *Pattern Recogn. Lett.*, 88(C):26–32.

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Chandramouli Shama Sastry and Sageev Oore. 2020. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support vector method for novelty detection. *Advances in neural information processing systems*, 12.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In

Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642.

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

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, volume 11856 of *Lecture Notes in Computer Science*, pages 194–206. Springer.

Yiyou Sun, Chuan Guo, and Yixuan Li. 2021. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 144–157.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. 2022. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 20827–20840. PMLR.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32.

Dennis Ulmer, Lotta Meijerink, and Giovanni Cinà. 2020. Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In *Machine Learning for Health*, pages 341–354. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3637–3645.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

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

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, VS@NAACL-HLT 2015, June 5, 2015, Denver, Colorado, USA*, pages 62–69. The Association for Computational Linguistics.

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

Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, WENXUAN PENG, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. 2022. OpenOOD: Benchmarking generalized out-of-distribution detection. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

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

Lam. 2022. A closer look at few-shot out-of-distribution intent detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 451–460.

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

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

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

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. Contrastive out-of-distribution detection for pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by

watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.