Enhancing Out-of-Distribution Detection in Natural Language Understanding via Implicit Layer Ensemble

Hyunsoo Cho[†], Choonghyun Park[†], Jaewook Kang[♮],
Kang Min Yoo[‡♮†], Taeuk Kim[§*], Sang-goo Lee[†]
[†]Seoul National University, [‡]NAVER AI Lab, [♮]NAVER CLOVA, [§]Hanyang University
{johyunsoo,pch330,sglee}@europa.snu.ac.kr
{jaewook.kang, kangmin.yoo}@navercorp.com
kimtaeuk@hanyang.ac.kr

Abstract

Out-of-distribution (OOD) detection aims to discern outliers from the intended data distribution, which is crucial to maintaining high reliability and a good user experience. Most recent studies in OOD detection utilize the information from a single representation that resides in the penultimate layer to determine whether the input is anomalous or not. Although such a method is straightforward, the potential of diverse information in the intermediate layers is overlooked. In this paper, we propose a novel framework based on contrastive learning that encourages intermediate features to learn layer-specialized representations and assembles them *implicitly* into a single representation to absorb rich information in the pre-trained language model. Extensive experiments in various intent classification and OOD datasets demonstrate that our approach is significantly more effective than other works. The source code for our model is available online.[1]

1 Introduction

Natural language understanding (NLU) in dialog systems, which often formalizes as a classification task to identify intentions behind user input, is a vital component as their decision propagates to the downstream pipelines. Numerous works have achieved immense success on sundry tasks (*e.g.,* intention classification, NLI, QA) reaching parity with human performance (Wang et al., 2019). Despite their success in many different benchmarks, neural models are known to be vulnerable to test inputs from an unknown distribution (Hendrycks and Gimpel, 2017; Hein et al., 2019), commonly referred to as outliers, since they depend strongly on the closed-world assumption (i.e., I.I.D assumption). Thus, out-of-distribution (OOD) detection (Aggarwal, 2017), which aims to discern outliers from the train distribution, is a essential research problem for ensuring a high-quality user experience and maintaining strong reliability as the systems in the wild encounter myriad unseen data ceaselessly.

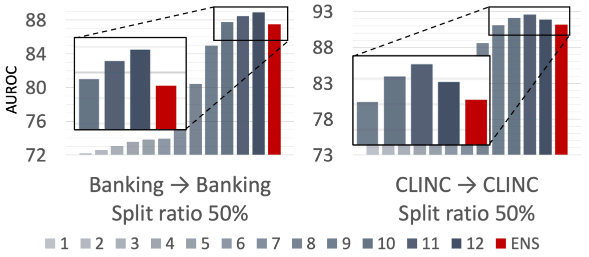


Figure 1: Layer-wise performances and their explicit ensemble (Shen et al., 2021) performance on BERT-base. Explicit ensemble often lead to worse AUROC (higher the better) than using a single well-performing layer. Detailed explanations about setting and baseline model are elaborated in Sec.4.2.1 and Sec.4.3 individually.

The most prevailing paradigm in OOD detection is to *extract* and *score*. Namely, it extracts the representation of the input from a neural model and passes it to a pre-defined scoring function. Then, the scoring function gauges the appropriateness of the input based on the extracted feature and decides whether the input is from the normal distribution. The most common rule of thumb for extracting representation from neural models is employing the the last layer, a simple and intuitive way to obtain a holistic representation, which is universally utilized in broad machine learning areas.

Meanwhile, previous studies (Tenney et al., 2019; Clark et al., 2019) revealed that the middle layers of the language model also conceal copious information. For instance, prior studies on language model probing suggest that syntactic linguistic knowledge is most prominent in the middle layers (Hewitt and Manning, 2019; Goldberg, 2019; Jawahar et al., 2019), and semantic knowledge in BERT is spread in all layers widely (Tenney et al., 2019). In this regard, leveraging intermediate lay-

[*] Corresponding author.

[1] https://github.com/HyunsooCho77/LaCL-official

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 783–798
December 7-11, 2022 ©2022 Association for Computational Linguistics

ers can lead to a better OOD detection performance, as they retain some complementary information to the last layer feature, which might be beneficial in discriminating outliers. Several studies (Shen et al., 2021; Sastry and Oore, 2020; Lee et al., 2018b) have shown empirical evidence that intermediate representations are indeed beneficial in detecting outliers. Precisely, they attempted to utilize middle layers via naïvely aggregating the individual result of every single intermediate feature ***explicitly***.

Although previous studies have shown the potential of intermediate layer representations in OOD detection, we confirmed that the aforementioned naïve ensemble scheme spawns several problems: (Fig. 1 illustrates OOD performance of the layer-wise and their explicit ensemble in two different datasets.) The first problem we observed is that the ensemble result (red bar) nor the last layer can not guarantee the best performance among the entire layer depending on the setting. Such a phenomenon raises the necessity for a more elaborate approach of deriving a more meaningful ensemble representation from various representations rather than a current simple summation or selecting a single layer. Secondly, even when this explicit ensemble gives a sound performance, it requires multiple computations of the scoring function by birth. Thus, explicit ensemble inevitably delays the detecting time, which is a critical shortcoming in OOD detection, as swift and precise decision-making is the cornerstone in this area.

To remedy the limitations of the explicit ensemble schemes, we propose a novel framework dubbed Layer-agnostic Contrastive Learning (LaCL). Our framework is inspired by the foundation of an ensemble, which seeks a more calibrated output by combining heterogeneous decisions from multiple models (Kuncheva and Whitaker, 2003; Gashler et al., 2008). Specifically, LaCL regards intermediate layers as independent decision-makers and assembles them into a single vector to yield a more accurate prediction: LaCL makes middle-layer representations richer and more diverse by injecting the advantage of contrastive learning (CL) into intermediate layers while suppressing inter-layer representations from being similar through additional regularization loss. Then, LaCL assembles them into a single ensemble representation ***implicitly*** to circumvent multiple computations of the scoring function.

We demonstrate the effectiveness of our approach in 9 different OOD scenarios where LaCL consistently surpasses other competitive works and their explicit ensemble performance by a significant margin. Moreover, we conducted an in-depth analysis of LaCL to elucidate its behavior in conjunction with our intuition.

2 Related Work

OOD detection. Methodologies in OOD detection can be divided into supervised (Hendrycks et al., 2019; Lee et al., 2018a; Dhamija et al., 2018) and unsupervised settings according to the presence of training data from OOD. Since the scope of OOD covers nigh infinite space, gathering the data in the whole OOD space is infeasible. For this realistic reason, the most recent OOD detection studies generally discriminate OOD input in an unsupervised manner, including this work. Numerous branches of machine learning tactics are employed for unsupervised OOD detection: generating pseudo-OOD data (Chen and Yu, 2021; Zheng et al., 2020), Bayesian methods (Malinin and Gales, 2018), self-supervised learning based approaches (Moon et al., 2021; Manolache et al., 2021; Li et al., 2021; Zhou et al., 2021; Zeng et al., 2021; Zhan et al., 2021), and novel scoring functions which measure the uncertainty of the given input (Hendrycks and Gimpel, 2017; Lee et al., 2018b; Liu et al., 2020; Tack et al., 2020).

Contrastive learning & OOD detection. Among the numerous approaches mentioned, contrastive learning (CL) based methods (Chen et al., 2020; Zbontar et al., 2021; Grill et al., 2020) are recently spurring predominant interest in OOD detection research. The superiority of CL in OOD detection comes from the fact that it can guide a neural model to learn semantic similarity within data instances. Such property is also precious for unsupervised OOD detection, as there is no accessible clue regarding outliers or abnormal distribution. Despite its potential, CL has been utilized in the computer vision field (Cho et al., 2021; Sehwag et al., 2021; Tack et al., 2020; Winkens et al., 2020) in the early works due to its high reliance on data augmentation. However, now it is also widely used in various NLP applications with the help of recent progress (Li et al., 2021; Liu et al., 2021; Kim et al., 2021; Carlsson et al., 2020; Gao et al., 2021; Sennrich et al., 2016). Specifically, Li et al. (2021) verified that CL is also helpful in the NLP field, and Zhou et al. (2021); Zeng et al. (2021) redesigned the contrastive-learning objective into a more appropriate form for OOD detection.

Potential of intermediate representation. The leading driver of the recent upheaval in NLP is the pre-trained language model (PLM), such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018), which trains a large-scale dataset on a transformer-based architecture (Vaswani et al., 2017). Numerous studies attempted to reveal the role and characteristics of each layer in PLMs and verified that diverse information is concealed in the middle layer, which is now a pervasive notion in the machine learning community. For instance, Tenney et al. (2019) showed that the different layers of the BERT network could resolve syntactic and semantic structure within a sentence. Clark et al. (2019) proposed an attention-based probing classifier leveraging syntactic information in the middle layer of BERT. Several studies (Shen et al., 2021; Sastry and Oore, 2020; Lee et al., 2018b) have shown the potential of intermediate representations in OOD detection by explicitly aggregating the individual result of every single intermediate feature.

3 Layer-agnostic Contrastive Learning

3.1 Intuition

The prime objective of our framework is to assemble rich information in the entire layers into a single ensemble representation to derive a more reliable decision. Inspired by the foundation of ensemble learning, which seeks better predictive performance by combining the predictions from multiple models, we regard each intermediate layer as an independent model (or decision maker). To make each layer a better decision-maker, LaCL injects a sound representation learning signal (i.e., supervised contrastive learning) to the entire layer by training objective function in a layer-agnostic manner to engage every layer more directly. Additionally, we propose correlation regularization loss (CR loss) which decorrelates a pair of strongly correlated adjacent representations to encourage each layer to learn layer-specialized representations from complementary information of each layer. Then, the global compression layer (GCL) ***implicitly*** assembles various features in each layer into a single calibrated ensemble representation . In the following subsections, we explain the components of our model in detail.

3.2 Supervised Contrastive Learning

Supervised contrastive learning (SCL) is a supervised variant of vanilla contrastive learning, which employs label information of the input to group samples into known classes more tightly. Thus, SCL can learn *data-label* relationships as well as *data-data* relationships as in CL.

In SCL, each batch $\mathcal{B} = \{(\boldsymbol{x}_b, y_b)\}_{b=1}^{|\mathcal{B}|}$ in the dataset, where $\boldsymbol{x}_b, y_b$ denotes a sentence and a label for index b respectively, generates an augmented batch $\bar{\mathcal{B}} = \{(\bar{\boldsymbol{x}}_b, \bar{y}_b)\}_{b=1}^{|\bar{\mathcal{B}}|}$, where labels of augmented views are preserved as the original one. The augmented batch $\bar{\mathcal{B}}$ consists of two augmented input; $\bar{\boldsymbol{x}}_{2b-1} = t_1(\boldsymbol{x}_b)$ and $\bar{\boldsymbol{x}}_{2b} = t_2(\boldsymbol{x}_b)$, where t_1, t_2 indicate data augmentation functions specified in Section 3.6. Then, $(\bar{\boldsymbol{x}}_{2b-1}, \bar{\boldsymbol{x}}_{2b})$ are passed through PLM and projector, generating latent vectors $(\boldsymbol{z}_{2b-1}, \boldsymbol{z}_{2b})$ that are utilized to calculate the supervised contrastive loss:

$$\mathcal{L}_{\text{SCL}} = -\log \sum_{j \in P(i)} \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_j / \tau)}{\sum_{k=1}^{|\bar{\mathcal{B}}|} \mathbb{1}_{[k \neq i]} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_k / \tau)}, \tag{1}$$

where $P(i) = \{p \in \mathcal{B} : \bar{y}_j = \bar{y}_i\}$ is the set of indices of all positives in the augmented batch with query index i and τ represents temperature hyperparameter.

3.3 Global Compression Layer

The global compression layer (GCL) is a two-layer MLP that is directly connected to entire layers to assemble intermediate representations into a single representation $\boldsymbol{z}$. GCL can be viewed as a particular type of projection head in contrastive learning. By linking the projection head to the entire layer, GCL facilitates layer-agnostic training to engage every middle layer in a training objective directly.

The process of extracting the final latent vector $\boldsymbol{z}$ with GCL is as follows: (The batch index term b is omitted for brevity from now.)

First, each layer l ($l \in |L|$, where $|L|$ refers to the cardinality of the layers) in PLM, outputs token embeddings $\boldsymbol{H}^l = [\boldsymbol{h}_1^l, \boldsymbol{h}_2^l, \cdots, \boldsymbol{h}_{len(\boldsymbol{x})}^l]$ for sentence $\boldsymbol{x}$. Then we combine token embeddings $\boldsymbol{H}^l$ into a single vector $\boldsymbol{h}^l = pool(\boldsymbol{H}^l)$ by applying the pooling function (i.e., mean pooling). Lastly, GCL receives the pooled token embedding of each layer $\boldsymbol{h}^l$ (where, $\boldsymbol{h}^l \in R^{|D|}$) as an input and outputs compact low-dimensional representation $\boldsymbol{c}^l$ (where, $\boldsymbol{c}^l \in R^{|D|/|L|}$). And we concatenate all compact

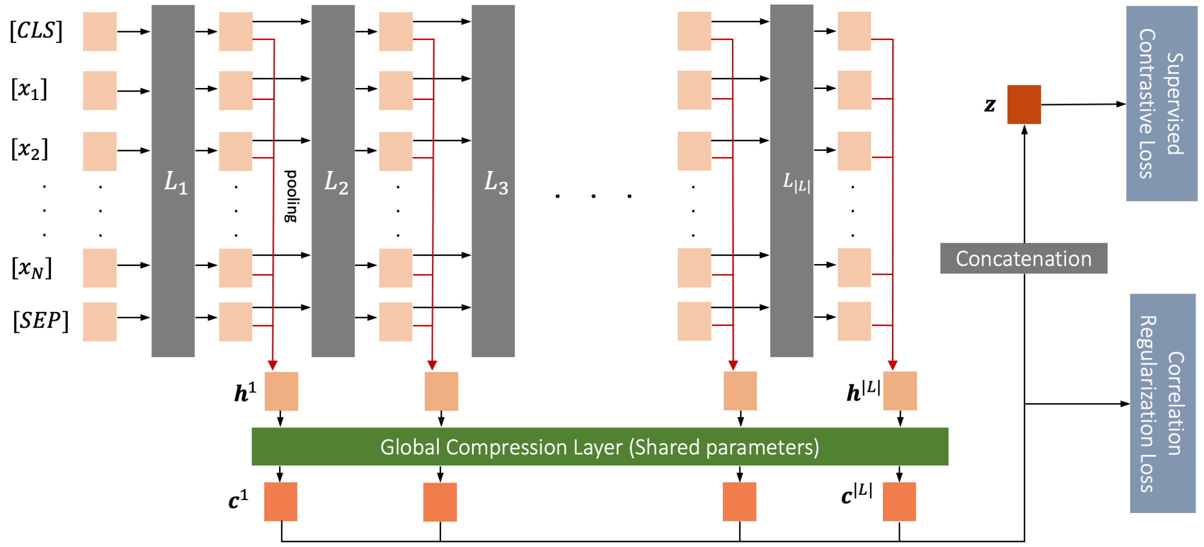

[CLS]
[x_1]
[x_2]
[x_N]
[SEP]
L_1
L_2
L_3
L_{|L|}
pooling
h^1
h^{|L|}
Global Compression Layer (Shared parameters)
c^1
c^{|L|}
z
Concatenation
Supervised Contrastive Loss
Correlation Regularization Loss

Figure 2: Overall structure of Layer-agnostic Contrastive Learning (LaCL). The global compression layer trains the SCL loss in a *layer-agnostic* manner by engaging entire layers in the CL task. And the correlation regularization (CR) loss decorrelates each intermediate layer to avoid ovelapping information between each layer.

representations $\boldsymbol{c}^l$ to generate a single sentence representation $\boldsymbol{z}$ from x:

$$\boldsymbol{z}(\boldsymbol{x}) = [\boldsymbol{c}^1 \oplus \boldsymbol{c}^2 \oplus \boldsymbol{c}^3 \oplus \cdots \oplus \boldsymbol{c}^{|L|}], \qquad (2)$$

where $\oplus$ indicates concatenation and $\boldsymbol{z} \in R^{|D|}$.

LaCL trains the SCL loss with the final representation from GCL $\boldsymbol{z}$, which inheres information from entire layers.

3.4 Correlation Regularization Loss

The correlation regularization (CR) loss restrains a pair of features from each adjacent layer from being similar, following the intuition of an ensemble where its performance boost springs from various decisions (Kuncheva and Whitaker, 2003; Gashler et al., 2008). Specifically, it encourages adjacent layers to activate different dimensions given the same input. First, we define the correlation in the dimension d of the adjacent layer (l and $l+1$) as follows:

$$cor^{\,d}_{(l,l+1)} = \frac{\sum_b \boldsymbol{c}^l_{b,d} \cdot \boldsymbol{c}^{l+1}_{b,d}}{\sqrt{\sum_b (\boldsymbol{c}^l_{b,d})^2}\sqrt{\sum_b (\boldsymbol{c}^{l+1}_{b,d})^2}}. \qquad (3)$$

where d indicates the index of hidden embedding dimension ($d \in |D|/|L|$, where $\boldsymbol{c}^l \in R^{|D|/|L|}$) and b refers to a data index of the augmented batch $\bar{\mathcal{B}}$.

Then, the CR loss selects a strongly correlated dimension set S by picking the dimensions that exceed the pre-set margin value m and decorrelates set S iterating over every adjacent layer:

$$S = \{d \in |D| : cor^{\,d}_{(l,l+1)} \geq m\}$$

$$\mathcal{L}_{\text{CR}} = \sum_l \sum_{d \in S} cor^{\,d}_{(l,l+1)}. \qquad (4)$$

Finally, the overall loss term for LaCL can be described as follows:

$$\mathcal{L}_{\text{LaCL}} = \mathcal{L}_{\text{SCL}} + \lambda_1 \mathcal{L}_{\text{CR}}, \qquad (5)$$

where λ_1 denote weights for CR loss.

3.5 Classification & OOD Scoring

Since there is no task-specific final layer (i.e., classification layer for cross-entropy loss) in LaCL, classification and anomaly detection are conducted via a cosine similarity scoring function (Tack et al., 2020). Employing the cosine similarity scoring function in LaCL is straightforward and shows good compatibility, as the model trained with contrastive learning can measure meaningful cosine similarity between data instances.

For input $\boldsymbol{x}$, we first extract the implicit ensemble representation $\boldsymbol{z}(\boldsymbol{x})$ and find the nearest neighbor instance $\boldsymbol{x}_{nn}$, i.e., $\max_{nn} \text{sim}(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x}_{nn}))$, from the training dataset. Then we classify label of $\boldsymbol{x}$ as the label of the nearest neighbor y_{nn}. And for the OOD detection, we use the similarity between input and its nearest neighbor as follows:

$$\text{Score}(\boldsymbol{x}) = \text{sim}(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x}_{nn})) \qquad (6)$$

Finally, we decide whether the input $\boldsymbol{x}$ is outlier or not through following the binary decision function I_δ:

$$I_\delta(\boldsymbol{x}) = \begin{cases} \text{IND} & \text{Score}(\boldsymbol{x}) \geq \delta \\ \text{OOD} & \text{Score}(\boldsymbol{x}) < \delta, \end{cases} \tag{7}$$

where δ denotes anomaly threshold, usually obtained from a score of the training instance which is in the boundary of the pre-set *true positive rate*.

3.6 Augmentation for Contrastive Learning

Augmentation is a crucial factor in CL that directly influence the model performance. To find the most effective data augmentation for OOD, we carefully select six data augmentation tactics for contrastive learning: back-translation (BT) (Li et al., 2021), dropout (DO) (Gao et al., 2021), token cutoff (Yan et al., 2021; Shen et al., 2020), random span masking (RSM) (Liu et al., 2021), and token shuffling (Lee et al., 2020). As our final data augmentation tactics, we greedily combined two best-performing augmentations, i.e., BT and RSM.

Instance 1 (t_1): raw data + RSM + DO

Instance 2 (t_2): BT + RSM + DO Note that DO is always applied by default unless the dropout probability is specified to 0 manually since it utilizes a dropout layer inside the transformer (Vaswani et al., 2017). We explain each augmentation and report their performance in the Appendix A.

4 Experiments

4.1 Implementation Details

In the following experiments, we adopt BERT-base (Devlin et al., 2019) as a backbone of our network. We fixed the dimension of the first layer in GCL to 1024 and the dimension of the second layer to $64 = 768/(num\_layers)$ so that the dimension of the concatenated vector $\boldsymbol{z}$ is 768 (BERT-base embedding dimension). We used mean pooling as a token embedding pooling function, set temperature τ to 0.05, CR loss weight λ_2 to 1, and margin m in CR loss to 0.5. Moreover, we set the batch size to 128 and used AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate 1e-5 with a cosine annealing scheduler.

4.2 Dataset and Metrics

4.2.1 Dataset.

We utilized CLINC150 (Larson et al., 2019), Banking77 (Casanueva et al., 2020), and Snips (Coucke et al., 2018) datasets for our experiments, which are commonly used in OOD detection literature. (The Appendix B covers statistics, description, and the detailed rationale behind our dataset selection.) Utilizing the selected dataset, we measure OOD performance in 9 different scenarios that can be categorized into the following two settings that are widely used in OOD detection:

- **Close-OOD setting (spliting dataset)** refers to a setting when the test distribution (OOD distribution) is *close* to the train distribution. Usually, close-OOD setting is simulated by partitioning one dataset into 2 disjoint datasets (i.e., IND / OOD dataset) based on the class label. Since the IND and OOD datasets originated from the equivalent dataset, they share similar distributions and properties, making the task more demanding. In our experiments, we randomly partitioned the class labels in each dataset with three different ratios (25%, 50%, and 75%), following the validation sets-up in previous works (Shu et al., 2017; Fei and Liu, 2016; Lin and Xu, 2019).
- **Far-OOD setting (distinct dataset)** refers to a setting when the test distribution (OOD distribution) is far from the IND train distribution. So far-OOD is relatively easy to discern test samples from the normal distribution. Usually, far-OOD setting is simulated by regarding the disjoint dataset as a test dataset (OOD dataset). i.e., CLINC150 (IND) → Banking77 (OOD) or Snips (OOD). In some scenarios, we verified that some intents belong to both IND and OOD, so we manually removed overlapping intents before training. (Details about removed intents in each scenario are in the Appendix B.2) We also categorize CLINC150 (OOD) → CLINC150 OOD split (OOD)[2] as far-OOD, since previous work (Zhang et al., 2022) manually confirmed that the distribution of CLINC OOD split is highly unrelated to CLINC train split.

4.2.2 Metrics.

To evaluate IND performance, we measured the classification accuracy. And for OOD metrics, we adopt two metrics that are commonly used in recent OOD detection literature:

- **FPR@95.** The false-positive rate at the true-positive rate of 95% (FPR@95) measures the probability of classifying OOD input as IND input when the true-positive rate is 95%.

[2] CLINC150 dataset has an internal OOD split dataset to measure the OOD performance.

BERT-base	IND : CLINC split (50%) → OOD : CLINC split (50%)								
	ACC	Cosine-single		Cosine-ENS		Mahalanobis-single		Mahalanobis-ENS	
		FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑
Baseline	96.74±0.36	38.97±2.88	92.10±0.53	**37.18**±0.46	**92.39**±0.27	39.33±1.31	91.74±0.22	40.14±0.33	91.05±0.39
DOC (Shu et al., 2017)	95.68±0.32	48.19±1.72	89.81±0.40	**42.11**±1.22	**90.79**±0.26	48.02±1.24	89.61±0.43	46.87±1.45	89.45±0.35
ConSERT (Yan et al., 2021)	97.42±0.26	35.68±1.31	**93.36**±0.45	**31.56**±1.42	93.08±0.26	34.35±1.66	93.34±0.51	34.40±1.01	92.23±0.45
SimCSE (Gao et al., 2021)	96.79±0.26	40.65±0.64	92.11±0.45	**36.05**±1.16	**92.32**±0.07	39.70±0.48	92.27±0.45	38.92±1.62	91.02±0.26
MirrorBERT (Liu et al., 2021)	97.60±0.30	34.22±0.92	93.75±0.28	**30.49**±1.49	93.38±0.22	33.86±1.63	**93.82**±0.29	33.92±1.17	92.77±0.31
Li et al. (2021)	97.31±0.20	36.10±1.59	**93.02**±0.44	**32.84**±1.79	92.82±0.33	35.14±1.46	92.98±0.51	36.38±1.34	91.64±0.62
Zhou et al. (2021)	96.56±0.24	36.10±2.43	93.15±0.53	39.22±1.01	92.46±1.11	**35.62**±3.34	**93.21**±0.39	40.34±1.39	92.02±0.94
Zeng et al. (2021)	96.47±0.44	45.30±3.07	**90.01**±0.42	-	-	**45.09**±2.15	89.34±0.39	-	-
LaCL (ours)	98.04±0.11	**26.59**±1.27	**94.93**±0.15	30.81±1.62	93.77±0.21	28.03±1.15	94.49±0.53	37.40±1.51	92.07±0.26
	IND : Banking split (50%) → OOD : Banking split (50%)								
Baseline	94.61±0.74	55.56±1.80	88.64±0.31	**52.52**±2.78	**90.57**±0.29	56.22±2.19	88.62±0.18	60.49±4.75	87.16±0.70
DOC (Shu et al., 2017)	94.50±0.19	59.49±1.10	86.98±0.60	**52.48**±3.45	**89.66**±0.37	60.54±0.64	86.75±0.83	57.69±1.54	87.51±0.55
ConSERT (Yan et al., 2021)	94.91±0.20	50.09±4.34	90.18±0.35	**46.33**±2.02	**91.30**±0.10	53.08±3.61	89.75±0.92	58.82±1.03	88.23±0.37
SimCSE (Gao et al., 2021)	94.83±0.47	54.23±2.03	90.27±0.76	**46.01**±1.90	**91.34**±0.32	52.78±2.64	90.08±0.82	59.72±2.99	87.88±0.79
MirrorBERT (Liu et al., 2021)	95.29±0.47	48.55±0.81	90.81±0.26	**43.67**±0.79	**91.58**±0.20	48.70±2.25	90.53±0.21	55.75±1.64	88.59±0.06
Li et al. (2021)	95.42±0.36	46.33±2.48	91.33±0.22	**42.91**±1.19	**91.95**±0.23	45.68±2.66	91.12±0.15	57.48±1.16	89.02±0.07
Zhou et al. (2021)	93.82±0.69	**52.86**±4.07	**89.43**±0.18	55.28±2.11	88.94±1.47	55.15±2.04	88.86±0.21	58.67±2.39	87.60±1.32
Zeng et al. (2021)	93.37±0.09	**56.91**±2.61	**83.12**±0.88	-	-	57.37±1.67	82.50±0.90	-	-
LaCL (ours)	95.51±0.27	**35.71**±0.61	**92.86**±0.16	42.67±1.67	91.58±0.11	47.73±7.12	89.88±1.21	69.00±0.60	83.66±0.94

Table 1: IND / OOD performance of each model 3 close-OOD settings. The best performance in each method is indicated in **bold** and the global best is underlined.

- **AUROC.** The area under the receiver operating characteristic curve (AUROC) is a threshold-free metric that indicates the ability of the model to discriminate outliers from IND samples.

4.3 Competing Methods

Recent OOD detection methods can be divided into scoring function and model training methods. We compare LaCL with their combinations to investigate the effectiveness in a holistic view.

Scoring functions:

- **Mahalanobis distance** discerns abnormal input via class-wise density estimation assuming the representation follows the multivariate normal distributions (Lee et al., 2018b). It is a multi-dimensional generalization of quantifying *how many standard deviations away from the mean of the distribution.* We also cover the explicit ensemble of the Mahalanobis (Shen et al., 2021), which is a simple aggregation of the Mahalanobis distance (D) of intermediate representations:

$$\mathrm{D}_{ens}(x) = \mathrm{D}(f^{|L|(\boldsymbol{x})}) + \sum_{1\leq l<|L|} \mathrm{D}(tanh(f^{l}(\boldsymbol{x}))) \tag{8}$$

Notably, they place the nonlinear $tanh$ layer to map the features of each transformer layer.

- **Cosine similarity** determines outliers by utilizing the similarity between the nearest neighbor of the known instance (usually from the training dataset) and the inferring input. Sec. 3.5 elaborates the details of the cosine similarity scoring function. We also cover an explicit ensemble version of the cosine scoring function, which determines OOD with an aggregation of cosine similarity of intermediate representations analogous to Eq. 8 but without $tanh$ function in the last term.

Training methods: We set a cross-entropy loss trained model and a sigmoid based 1-vs-rest classifier (Shu et al., 2017) as a baseline model. Additionally, we compare our method with 6 recent CL based methods (Gao et al., 2021; Liu et al., 2021; Yan et al., 2021; Li et al., 2021; Zhang et al., 2022; Zhou et al., 2021): Precisely, Gao et al. (2021); Liu et al. (2021); Yan et al. (2021) [3] suggest a general CL framework, while Li et al. (2021); Zhang et al. (2022); Zhou et al. (2021) introduce CL for OOD detection, which redesigns the loss function to maximize the discrepancy between IND and OOD.

For unsupervised CL methods, we train them with the cross-entropy loss additionally to give a signal about training distribution as in OOD specific frameworks. We extract the mean-pooled representation of the last layer features for all methods and pass it to a scoring function. On the other hand, LaCL exploits an implicit ensemble representation $\boldsymbol{z}$ from GCL.

4.4 Main Results

This section reports the performance of LaCL with other competing methods in two different settings. Tab. 1 summarizes IND and OOD performance in close-OOD scenarios when the split ratio is 50% and Tab. 2 summarizes IND and OOD performance

[3] Explanations of each method is in the Appendix A.

BERT-base	IND : CLINC → OOD : CLINC internal OOD								
	ACC ↑	Cosine-single		Cosine-ENS		Mahalanobis-single		Mahalanobis-ENS	
	(IND)	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑
Baseline	95.62±0.39	12.13±1.63	97.50±0.14	**10.20**±0.53	**97.84**±0.04	12.67±0.83	97.32±0.22	11.23±0.40	97.61±0.09
Shen et al. (2021)[†]	96.66	-	-	-	-	10.88	97.43	**10.12**	**97.77**
DOC (Shu et al., 2017)	94.69±0.48	19.23±1.34	96.63±0.12	**11.47**±1.59	**97.62**±0.05	19.07±1.07	96.65±0.11	16.73±0.67	97.07±0.13
ConSERT (Yan et al., 2021)	96.21±0.09	11.00±0.72	97.61±0.10	**8.37**±0.15	**98.00**±0.05	10.33±0.76	97.67±0.12	9.60±0.46	97.78±0.13
SimCSE (Gao et al., 2021)	95.80±0.30	12.50±0.10	97.60±0.14	**8.80**±0.20	**97.96**±0.02	11.63±0.21	97.65±0.10	10.50±0.50	97.67±0.03
MirrorBERT (Liu et al., 2021)	96.50±0.15	10.33±0.61	97.78±0.04	**8.40**±0.17	**98.09**±0.06	9.60±0.17	97.82±0.03	8.67±0.40	97.94±0.04
Li et al. (2021)	96.08±0.19	11.03±0.65	97.68±0.12	**8.90**±0.36	**98.01**±0.16	10.27±0.55	97.70±0.11	9.33±0.64	97.81±0.14
Zhou et al. (2021)	95.28±0.27	10.93±0.74	97.65±0.15	**9.60**±0.28	**97.67**±0.18	10.43±0.90	97.66±0.18	10.80±0.71	97.51±0.37
Zeng et al. (2021)[⋆]	94.58±0.58	**19.87**±1.51	**96.43**±0.18	-	-	23.40±1.97	95.75±0.20	-	-
LaCL (ours)	96.96±0.39	**6.67**±0.51	**98.27**±0.16	7.43±0.06	98.15±0.07	8.30±0.61	98.00±0.17	12.33±0.12	97.31±0.11
	IND : CLINC → OOD : Banking								
Baseline	96.67±0.13	13.10±1.61	97.42±0.17	**10.69**±0.77	**97.62**±0.11	13.85±1.66	97.17±0.10	12.28±0.48	97.54±0.06
ConSERT (Yan et al., 2021)	96.74±0.33	10.71±2.58	97.83±0.36	9.96±1.73	97.64±0.18	**9.53**±2.44	**98.02**±0.33	9.59±2.35	97.78±0.26
SimCSE (Gao et al., 2021)	96.55±0.07	12.19±2.14	97.69±0.38	**10.01**±2.17	97.66±0.33	10.94±2.50	**97.79**±0.42	10.80±2.69	97.61±0.41
MirrorBERT (Liu et al., 2021)	97.00±0.10	9.29±1.19	98.04±0.28	9.63±1.37	97.76±0.18	**8.47**±1.42	**98.14**±0.27	9.75±0.88	97.90±0.23
DOC (Shu et al., 2017)	95.32±0.32	19.55±3.25	96.75±0.51	**13.45**±2.51	**97.37**±0.36	18.27±3.77	96.87±0.50	16.74±3.95	97.12±0.59
Li et al. (2021)	96.58±0.10	12.09±2.12	97.68±0.30	**9.92**±2.05	97.62±0.30	10.34±1.62	**97.87**±0.22	10.63±1.87	97.62±0.25
Zhou et al. (2021)	96.31±0.34	11.08±1.00	97.79±0.12	10.15±1.40	97.86±0.32	**8.40**±1.53	98.09±0.09	8.67±2.53	**98.11**±0.46
Zeng et al. (2021)[⋆]	95.20±0.25	**22.39**±4.09	**95.87**±0.43	-	-	23.06±3.32	95.64±0.38	-	-
LaCL (ours)	96.90±0.49	**4.86**±0.15	**98.57**±0.06	9.05±1.09	97.79±0.12	12.19±4.72	97.51±0.67	11.23±0.96	97.44±0.14
	IND : CLINC → OOD : Snips								
Baseline	95.83±0.08	27.10±1.44	96.11±0.03	**11.54**±1.08	97.65±0.16	20.33±0.98	96.68±0.19	11.54±0.33	**97.82**±0.12
DOC (Shu et al., 2017)	94.34±0.21	29.08±3.75	95.71±0.59	**18.46**±4.30	**96.86**±0.62	27.29±3.23	95.99±0.52	22.64±3.06	96.57±0.61
ConSERT (Yan et al., 2021)	96.14±0.24	18.20±1.87	97.08±0.25	**11.10**±0.87	**98.00**±0.15	15.93±1.81	97.42±0.26	12.97±1.15	97.92±0.18
SimCSE (Gao et al., 2021)	95.80±0.28	20.99±3.85	96.84±0.51	10.81±2.64	97.94±0.34	15.20±4.34	97.46±0.34	**10.59**±2.77	**98.04**±0.22
MirrorBERT (Liu et al., 2021)	96.57±0.21	18.17±2.64	97.14±0.37	**10.07**±1.34	98.04±0.31	14.25±2.76	97.56±0.38	10.81±1.89	**98.10**±0.24
Li et al. (2021)	96.06±0.24	16.96±2.53	97.24±0.17	**9.85**±0.94	98.09±0.22	13.19±2.02	97.62±0.22	10.26±1.77	**98.11**±0.19
Zhou et al. (2021)	95.09±0.43	19.20±3.20	96.87±0.39	19.01±3.85	97.00±0.45	**13.59**±3.47	97.57±0.34	14.39±2.85	**97.59**±0.31
Zeng et al. (2021)[⋆]	93.72±0.16	**28.77**±1.09	**94.75**±0.13	-	-	29.43±1.36	94.48±0.21	-	-
LaCL (ours)	96.62±0.45	**8.06**±1.59	98.24±0.17	8.17±0.63	**98.40**±0.08	11.06±0.92	98.02±0.11	11.36±0.64	98.11±0.05

[⋆] freezes the parameters of BERT, so we omitted ensemble evaluation. [†] performance report from original paper.

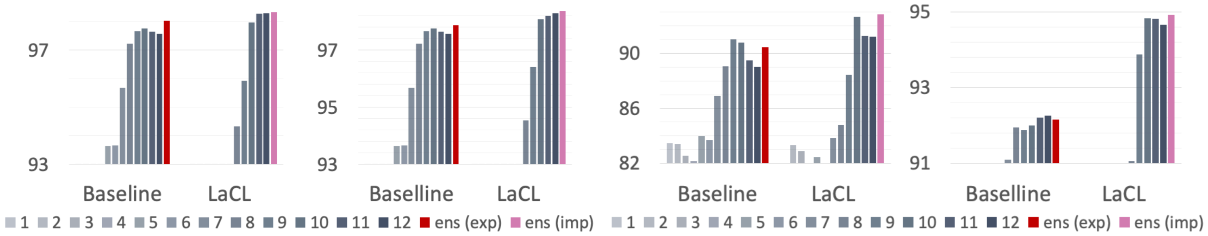
Table 2: IND / OOD performance of each model 3 far-OOD settings. The best performance in each method is indicated in **bold** and the global best is underlined.

in three far-OOD scenarios. (Performance report with the remaining ratios, i.e., 25%, and 75%, are in the Appendix C.) We report the average and standard deviation of 5 trials as a model performance for reproducibility.

From the results, we verified that LaCL with a cosine scoring (single) function consistently surpasses other methods significantly. We also confirmed that most methods (excluding LaCL) exhibit better performance with the explicit ensemble methods, indicating the potential of intermediate representations in OOD detection, as suggested in past studies (Shen et al., 2021; Sastry and Oore, 2020; Lee et al., 2018b). However, the performance of LaCL degrades with the explicit ensemble evaluations, proving that our ensemble method can gather more distinctive and calibrated information from entire layers than the naïve aggregation, and the explicit ensemble only acts as noise. It is also worth noticing that LaCL shows good compatibility with cosine evaluation than the Mahalanobis evaluation since the Mahalanobis evaluation assumes that the extracted representations follow a Gaussian distribution. The following condition holds when the model is trained with cross-entropy loss, as they can be viewed as a generative classifier (Lee et al., 2018b). However, LaCL does not utilize cross-entropy loss, and the mentioned assumption is hardly met. Lastly, cosine ensemble evaluation tends to perform better than the Mahalanobis ensemble (Shen et al., 2021) counterpart in general. We conjecture that aggregating each result into a single one is more difficult in the Mahalanobis ensemble, as the Mahalanobis distance is not a normalized score (ranging $-\infty$ to ∞) while cosine is normalized (ranging -1 to 1). To conclude, we demonstrate that our model can extract elaborate ensemble representation, which yields the highest performance in various scenarios without multiple computations of the scoring function.

5 Analysis

In this section, we conduct supplementary experiments on LaCL to analyze our framework in-depth to elucidate its behavior.



(a) Far-OOD.
CLINC → CLINC (OOD), CLINC → Banking

(b) Close-OOD.
Banking (ratio 50%), CLINC (ratio 50%)

Figure 3: Layer-wise AUROC score of baseline and LaCL with cosine scoring function. Explicit ensemble (baseline) tends to work well in relatively easy setting (far-OOD), while it yields worse performance than best performing single representation in harsh conditions (close-OOD). Implicit ensemble representation from LaCL outperforms other layers consistently.

5.1 Layer-wise Performance

Although our model outperforms other methods, it is unclear whether LaCL can well-assemble the information in the intermediate representations, analogous to our initial intuition. In an attempt to give answer this question, we scrutinize the layer-wise performance of LaCL and the baseline model. Fig. 3 summarizes the layer-wise AUROC score of LaCL and baseline in far-OOD and close-OOD settings. While higher layers tend to exhibit better performance, it is not always the case. Speaking otherwise, the last layer does not always guarantee the best performance among the upper layers. In this situation, the explicit ensemble of the baseline model conditionally shows performance gain. Namely, in a far-OOD setting (Fig. 3a), the ensemble representation displays substantial performance gain. In contrast, in a close-OOD setting (Fig. 3b), the ensemble representation often yields worse performance than the best-performing single layer. On the other hand, LaCL displays the best performances among other layers unconditionally, proving the capability of LaCL to absorb layer-specialized information of the entire layers properly.

5.2 Ablation study

We present ablations on LaCL to give intuition behind its behavior and justify our design choices.
Module ablations. We alter our model in several ways by removing some components in LaCL to test their independent impact. Tab. 3 summarizes component-wise ablations of our model in Banking 50% split setting, which is the harshest condition (lowest performance) from Tab. 2, 1. While our layer agnostic training (GCL) or regularization term (CR loss) does not statistically contribute to the accuracy compared to applying SCL alone, they substantially improve OOD performance.

Components	Acc ↑ (IND)	Cosine	
		AUROC ↑	FPR@95 ↓
Baseline	95.07	88.53	55.39
+ SCL	95.65	91.68	39.55
+ GCL	**95.72**	92.23	37.62
+ CR (LaCL)	95.66	**92.84**	**35.13**
LaCL (variant 1)	**95.72**	92.51	37.34
LaCL (variant 2)	95.52	92.55	38.71

Table 3: Ablation study on LaCL components and its variations on Banking split setting .

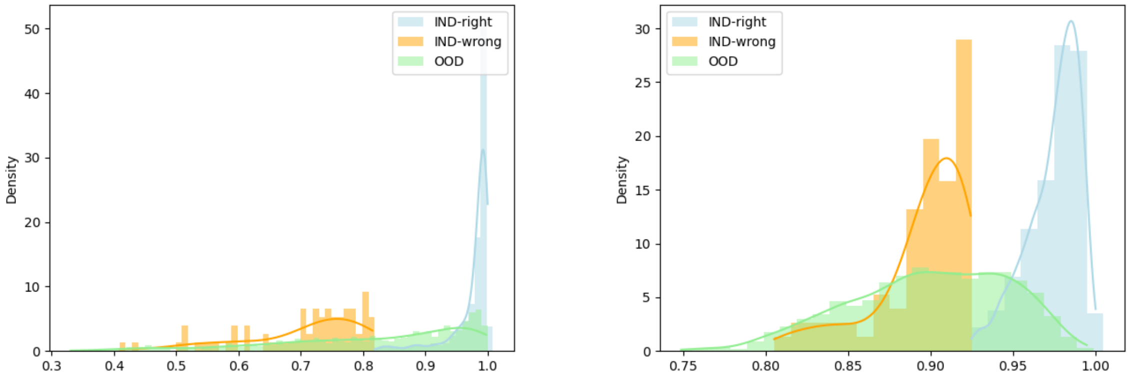
LaCL variants. From previous experiments (Sec. 5.1), we verified that the higher layers tend to yield better performance than the lower layers. So it is a reasonable conjecture that assembling only the upper layers may render better performance, assuming there is no meaningful information in the lower layers. Founded on this observation, we introduce two variants of LaCL: First variant (variant 1 in Tab. 3) utilize upper half layers $\boldsymbol{z}^*$ in the *inference*:

$$\boldsymbol{z}^* = [\boldsymbol{c}^{(|L|/2)} \oplus \boldsymbol{c}^{(|L|/2+1)} \oplus \cdots \oplus \boldsymbol{c}^{|L|}] \quad (9)$$

Furthermore, the second variant (variant 2) also utilizes the upper half layers $\boldsymbol{z}^*$; however, they disconnect the lower half layers with GCL when they train the model. To our surprise, we verified that LaCL outperforms the other two variants, indicating that the features from lower layers retain considerable meaningful information, regardless of their performance.

5.3 Distribution Visualization

In this section, we plot a histogram of our model and baseline model to visualize how each model



(a) Histogram of baseline model.

(b) Histogram of LaCL.

Figure 4: Histogram of LaCL and Baseline model trained on Banking split 50% setting.

OOD Input Text	Prediction
Where can i find cheap **rental** skis nearby	car_rental
Search up someone who **plays** in a movie	play_music
What **oil** is best for chicken	oil_change_how
Read **text**	text
What is harry's real **name**	change_user_name
Check **battery** health on this device	jump_start
Who invented the internet	who_made_you

Table 4: Examples of OOD samples misclassified as IND. The keywords that cause over-reliance are in **bold**.

forms the IND and OOD distribution. Fig. 4 illustrates the histogram with the cosine scoring function of LaCL and the baseline model trained on Banking split 50% setting. We regard inputs as OOD when the input score is lower than the threshold δ, where δ is a preset threshold when TPR is at 95%, as stipulated in FPR-95%. To our surprise, both models have the ability to discriminate IND-wrong (yellow line) from IND-right answer (blue line), meaning can output high uncertainty for inputs that are likely to be wrong. On the other hand, LaCL forms a much clearer decision boundary and measures more precisely predictive uncertainty for OOD inputs (green line).

5.4 Case Study

We also scrutinized a case study on misclassified OOD inputs to identify the shortcomings and limitations of our model. Tab. 4 summarizes some OOD inputs which LaCL misclassified as normal input along with their IND prediction class. In most cases, they include keywords or phrases that are highly relevant to the wrongly predicted intent, meaning they tend to learn some shortcuts (Geirhos et al., 2020) instead of capturing holistic context. Another notable observation is that LaCL is fragile to typos and non-standard language (*e.g.,* acronyms, slangs). More thorough explorations are in Appendix D.1.

6 Conclusion

In this study, we propose a novel framework called LaCL to improve OOD detection by leveraging intermediate representations. Our framework seeks a more calibrated output by combining layer-specialized representations from each layer via a layer-agnostic training scheme and novel regularization loss. Through extensive experiments and ablations, we have demonstrated the potential of intermediate representations in OOD detection and the effectiveness of our framework, which significantly outperforms other existing works.

7 Limitations

Currently, our model concatenates compressed representations to gather information from entire layers. Thereby, if the number of layers changes depending on the backbone, hyper-parameters of LaCL need to be manually optimized. Additionally, our methodology is a general-purpose methodology that can be applied to other tasks as well as OOD detection, but its utility has not been explored in other tasks. For future work, we will explore the compatibility of our framework to other tasks or areas (e.g., computer vision) and devise an approach to optimize the aforementioned hyper-parameters in an automated fashion.

8 Acknowledgements

This work was mainly supported by SNU-NAVER Hyperscale AI Center. Also, this work was partially supported by Artificial Intelligence Innovation Hub [No.2021-0-02068, Artificial Intelligence Institute, Seoul National University] and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No.2020-0-01373, Artificial Intelligence Graduate School Program, Hanyang University].

References

Charu C Aggarwal. 2017. An introduction to outlier analysis. In *Outlier analysis*, pages 1–34. Springer.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2020. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

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

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR.

Hyunsoo Cho, Jinseok Seol, and Sang-goo Lee. 2021. Masked contrastive learning for anomaly detection. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

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, NAACL*.

Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult. 2018. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems*.

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Mike Gashler, Christophe Giraud-Carrier, and Tony Martinez. 2008. Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 900–905. IEEE.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems, NeurIPS*.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. 2019. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–50.

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*.

Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. 2019. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics ACL*.

Ludmila I Kuncheva and Christopher J Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207.

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 EMNLP*.

Haejun Lee, Drew A. Hudson, Kangwook Lee, and Christopher D. Manning. 2020. SLM: learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*.

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*.

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*, pages 7167–7177.

Tian Li, Xiang Chen, Shanghang Zhang, Zhen Dong, and Kurt Keutzer. 2021. Cross-domain sentiment classification with contrastive learning and mutual information maximization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8203–8207. IEEE.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*.

Fangyu Liu, Ivan Vulic, Anna Korhonen, and Nigel Collier. 2021. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*.

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, NeurIPS 2020*.

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.

Andrey Malinin and Mark J. F. Gales. 2018. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems 31, NeurIPS 2018*.

Andrei Manolache, Florin Brad, and Elena Burceanu. 2021. DATE: detecting anomalies in text via self-supervision of transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*.

Seung Jun Moon, Sangwoo Mo, Kimin Lee, Jaeho Lee, and Jinwoo Shin. 2021. MASKER: masked keyword regularization for reliable text classification. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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

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.

Vikash Sehwag, Mung Chiang, and Prateek Mittal. 2021. SSD: A unified framework for self-supervised outlier detection. In *9th International Conference on Learning Representations, ICLR 2021*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for intent classification and out-of-domain detection in SLU. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics ACL*.

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*.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. 2020. CSI: novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems 33, NeurIPS 2020*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR*.

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, et al. 2020. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. 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*.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*. PMLR.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*.

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 ACL*.

Jianguo Zhang, Kazuma Hashimoto, Yao Wan, Zhiwei Liu, Ye Liu, Caiming Xiong, and Philip Yu. 2022. Are pre-trained transformers robust in intent classification? a missing ingredient in evaluation of out-of-scope intent detection. In *Proceedings of the 4th Workshop on NLP for Conversational AI*.

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

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, EMNLP*.

Appendix

A Data Augmentation Selection

This section provides in-depth explanations about the various data augmentation methods along with their performance in OOD detection.

Back-Translation is a method of translating a raw sentence into another language and then re-translating it back into the same language. Precisely, we translate raw sentence into german and re-translate it back into english utilizing 'transformer.wmt19.en-de.single_model', 'transformer.wmt19.de-en.single_model' from fairseq (Ott et al., 2019). In order to avoid the BT sentence from being completely identical to the original sentence, we generated top-5 sentences and sampled from them after checking the duplicates.

Dropout (Gao et al., 2021) utilize dropout layers in transformers (Vaswani et al., 2017) to extract stochastically different representation. Due to dropout layer, giving the same input to the same model yields slight different representation and dropout utilize those inputs as a augmentation. Note that, dropout is always applied by default.

Random Span Masking (RSM) first randomly select some span, i.e.,k continuous characters, in the input sequence. Then, they randomly replaced with [MASK] token. In general, RSM is apply in one instance of the two augmented instances, as it was proposed in the original MirrorBERT paper (Liu et al., 2021) In this paper, we additionally consider applying it on both side of a pair.

Token Shuffling Token shuffling randomly shuffles the order of the input tokens (positional embedding) in the input sequence.

Token Cutoff In token cutoff is a simple strategy that eliminates some input tokens randomly.

We investigate the effectiveness of the beforementioned augmentations in OOD detection to select our final data augmentation combination. Tab. A summarizes the results in Banking split 50% setting. For our final data augmentation, we greedily combined two best performing augmentations, i.e., BT and RSM, which showed best performance in OOD metrics.

Dataset	Avg length	# Domain	# Intent	# Class
CLINC150	8.31	10	15	150
Banking77	11.9	1	77	77
Snips	9.05	7	-	7

Table 5: Dataset statistics.

CLINC150	Snips
play_music	PlayMusic
update_playlist	AddToPlaylist
weather	GetWeather
confirm_reservation restaurant_reservation cancel_reservation accept_reservations	BookRestaurant

Table 6: Overlapping classes between CLINC and Snips dataset.

B Dataset

B.1 Dataset Selection and Details

In order to investigate the performance of our model in many different situations, we conduct experiments on intention classification datasets. Generally, intention classification classes are organized hierarchically, which often consist of domains (*e.g.,* banking, travel, reservation) and intents (*e.g.,* banking - transfer money, banking - check account) where one domain serve as a parent category of multiple intents. It is much demanding to distinguish unknown intent under equivalent domain than discerning unseen domain (Zhang et al., 2022), as the domain is a high-level concept. Considering the facts mentioned above, we carefully selected CLINC150 (Larson et al., 2019), Banking77 (Casanueva et al., 2020), and Snips (Coucke et al., 2018) datasets each comprises of distinct class hierarchy.

Specifically, CLINC150 dataset contains various domains and intents, so it is a favorable dataset to measure overall model performance. In the case of the Banking77 dataset, it consists of fine-grained 77 intents under a single banking domain. On the other hand, the Snips dataset comprises seven different domains, making each class relatively easy to discern. (See Tab. 5 for statistics about each dataset.)

B.2 Overlapping Intents

In far-OOD setting, we train the model with CLINC dataset and test with Snips or Banking dataset. While each dataset includes a variety of domains,

Method	Instance 1	Instance 2	AUROC	FPR-95	ACC
Baseline	-	-	88.75	57.18	95.07
Shuffle	Raw + DO	Raw + DO + Shuffle	89.05	51.54	94.41
Dropout	Raw + DO	Raw + DO	90.35	49.23	95.2
Token-Cutoff	Raw + DO	Raw + DO + TC	90.55	50.71	95.2
Back-Translation (BT)	Raw + DO	BT + DO	91.1	48.27	95
Random Span Masking (RSM)	Raw + DO	Raw + DO + RSM	90.91	48.85	**95.59**
RSM (pair)	Raw + DO + RSM	Raw + DO + RSM	91.75	43.89	95.26
RSM (pair) + BT	Raw + DO + RSM	BT + DO + RSM	**91.94**	**42.89**	95.26

Table 7: Data augmentaion results in Banking split 50% setting.

however, there is a potential overlap in between each datasets. We manually compared domains and intents in each dataset and removed overlapping classes, as there should be no overlap of domain between OOD test set and the train set. Specifically, *banking* and *credit_cards* domains in CLINC150 is similar to banking77, so we removed mentioned domain from CLINC before we train the model. Likewise, Snips also includes some intents that also occur in CLINC150, which are summarized in Tab 6. We removed 7 intents in CLINC dataset in Tab.6 when we utilize Snips dataset as an OOD dataset.

C Additional Experiments

C.1 Split setting with remaining ratios

In this experiment, we exhibit remaining experiments in split setting (ratio 25%, 75%) with the BERT backbone to the Table. The results resembles the tendency to the split ratio 50% in the main paper experiments. LaCL outperform other models significantly.

D Additional Qualitative results

D.1 Case study

In this section we elaborate detailed case study on LaCL trained and tested on CLINC150 setting. Following previous experiment in the paper, we regard inputs as OOD when the the input cosine score is lower than the threshold δ, where δ is a preset threshold when TPR is at 95%, as stipulated in FPR-95%.

To further investigate our error cases, we categorized the error cases into two classes: *OOD inputs, misclassified as IND*, and *IND inputs, misclassified as OOD*.

OOD inputs, misclassified as IND occurs when LaCL predicts a high confidence for OOD input. Example cases for this error are shown in Tab. 9. There are many controversial variation in this error in this case; however, they contain keywords or phrases that are highly relevant to the wrongly predicted IND intent, meaning they tend to learn some shortcuts (Geirhos et al., 2020) from the train set as mentioned in the paper. The fact that fine-tuned classifier learns some shortcuts from the train set is a well-known problem, and there are previous works (Moon et al., 2021). As a side note, few data were mislabeled, as can be seen in Tab. 9.

IND inputs, misclassified as OOD occurs when LaCL predicts a low confidence for IND input. Example cases for this error are summarized in Tab. 10. We sort the error cases into 3 groups: Misspelled words (typos), Nonstandard words (*e.g., acronyms, slangs*), and absence of keywords.

The examples in first error case occurs when the words heavily related to the intent are misspelled. Interesting part is that even though the model assigns low score to this type of errors, it predicts the true intent correctly. The second error type happens when nonstandard words appear. We believe that these errors are caused by the PLM not having semantics for those abnormal tokens. Lastly, final error case arises when the intent-specific words are absent in the sentence. Namely, LaCL suffers when the input sentence comprises the words that are not commonly used, although their semantic is roughly the same. This phenomenon is another example of the prominence of keyword over reliance. However, learning shortcut is natural phenomenon surmising the following example: The train data in the '*text*' intent, the word *text* appears in 96 out of 100 sentences. As a side note, few data were mislabeled similar to previous error cases.

BERT-base	IND : CLINC split (25%) → OOD : CLINC split (75%)								
	ACC	Cosine-single		Cosine-ENS		Mahalanobis-single		Mahalanobis-ENS	
		FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑	FPR@95 ↓	AUROC ↑
Baseline	98.11±0.05	27.59±0.20	94.49±0.03	**25.65**±0.49	94.46±0.02	26.76±0.47	**94.62**±0.08	27.36±1.18	93.83±0.01
DOC (Shu et al., 2017)	97.28±0.05	33.95±0.66	93.24±0.02	33.56±0.54	**93.58**±0.02	33.78±0.54	93.24±0.01	**32.67**±0.29	93.03±0.02
ConSERT (Yan et al., 2021)	99.00±0.14	23.22±0.06	95.01±0.06	23.57±0.70	94.73±0.04	22.79±0.80	**95.23**±0.06	**22.28**±1.30	94.50±0.10
SimCSE (Gao et al., 2021)	98.36±0.05	**25.43**±1.07	94.51±0.18	25.98±0.73	94.63±0.04	25.65±0.33	**94.78**±0.13	26.39±0.40	94.22±0.06
MirrorBERT (Liu et al., 2021)	99.11±0.05	22.39±0.52	95.49±0.34	22.99±1.38	95.10±0.19	21.57±0.75	**95.64**±0.33	**21.41**±1.25	95.00±0.27
Li et al. (2021)	98.80±0.48	25.58±0.28	94.75±0.05	**24.97**±0.69	94.69±0.05	25.93±0.56	**94.99**±0.10	26.08±1.17	94.38±0.15
Zhou et al. (2021)	97.53±1.35	**25.55**±1.13	**95.03**±0.11	26.15±2.01	94.57±0.35	26.98±1.85	94.86±0.21	27.22±1.05	94.05±0.16
Zeng et al. (2021)	98.61±0.24	**28.57**±2.13	**94.01**±0.60	-	-	29.48±2.41	93.72±0.50	-	-
LaCL (ours)	99.09±0.14	**18.03**±0.66	**96.31**±0.04	18.80±0.55	96.27±0.08	21.18±0.69	96.16±0.18	25.94±0.32	94.99±0.14
	IND : CLINC split (75%) → OOD : CLINC split (25%)								
Baseline	96.52±0.42	39.58±1.36	91.42±0.04	46.00±0.77	91.45±0.18	36.33±1.54	91.63±0.09	**35.33**±0.18	**93.24**±0.01
DOC (Shu et al., 2017)	94.97±0.14	51.03±0.75	89.47±0.25	50.69±0.96	89.80±0.24	50.86±0.80	89.12±0.31	**44.19**±0.76	**91.02**±0.26
ConSERT (Yan et al., 2021)	96.70±0.21	38.44±1.00	92.35±0.22	37.67±1.04	92.13±0.16	38.97±1.17	91.31±0.35	**35.42**±0.92	**92.88**±0.15
SimCSE (Gao et al., 2021)	96.09±0.29	40.20±1.57	91.97±0.41	40.05±1.80	91.85±0.45	37.97±0.67	91.00±0.39	**35.53**±0.99	**92.85**±0.23
MirrorBERT (Liu et al., 2021)	96.85±0.26	36.89±0.24	93.18±0.00	38.00±0.87	93.03±0.01	35.39±1.03	92.17±0.33	**32.69**±0.68	**93.38**±0.19
Li et al. (2021)	96.76±0.10	41.06±0.10	92.70±0.07	41.00±0.72	92.52±0.07	38.58±0.84	91.76±0.10	**34.61**±1.54	**93.04**±0.07
Zhou et al. (2021)	95.72±0.62	45.53±5.89	90.70±0.94	43.75±1.80	91.22±0.05	41.91±0.63	91.48±0.26	**41.75**±0.68	**91.94**±0.40
Zeng et al. (2021)	95.65±0.36	45.64±1.98	90.76±0.42	**44.61**±1.48	**91.62**±0.37	-	-	-	-
LaCL (ours)	97.38±0.16	**26.50**±0.33	**95.81** ±0.14	27.72±0.72	95.44±0.06	41.70±1.04	92.02±0.35	32.53±0.24	93.98±0.12
	IND : Banking split (25%) → OOD : Banking split (75%)								
Baseline	97.54±0.15	34.96±1.61	93.29±0.14	**34.84**±1.53	**93.83**±0.11	35.70±1.21	93.22±0.17	38.45±2.39	92.28±0.20
DOC (Shu et al., 2017)	97.15±0.08	38.44±0.62	91.85±0.22	**35.15**±2.82	**93.17**±0.22	39.15±0.56	92.06±0.07	37.84±0.64	91.92±0.27
ConSERT (Yan et al., 2021)	97.72±0.20	26.81±0.95	94.43±0.30	29.81±2.03	94.12±0.12	**26.37**±2.20	**94.48**±0.19	38.92±1.15	92.40±0.03
SimCSE (Gao et al., 2021)	97.24±0.60	28.29±1.83	**94.98**±0.12	30.07±2.41	94.25±0.17	**27.64**±1.72	94.87±0.17	45.23±1.52	91.87±0.15
MirrorBERT (Liu et al., 2021)	97.72±0.42	**27.11**±1.36	94.90±0.16	31.11±2.44	94.11±0.11	27.13±0.93	**95.05**±0.13	42.54±0.60	92.36±0.05
Li et al. (2021)	97.59±0.33	27.72±1.24	**95.15**±0.14	27.40±2.30	94.56±0.10	**26.61**±0.07	95.15±0.09	38.16±1.89	92.99±0.06
Zhou et al. (2021)	95.96±2.10	**32.05**±3.26	**92.80**±0.49	34.34±0.84	92.66±0.70	32.97±1.89	92.65±0.87	36.13±0.68	92.38±0.40
Zeng et al. (2021)	95.09±2.92	**44.71**±6.22	**89.93**±2.38	-	-	47.28±9.02	89.24±3.02	-	-
LaCL (ours)	98.03±0.00	**23.45**±2.05	**95.50**±0.05	29.59±2.43	94.22±0.20	35.92±2.67	93.56±0.48	48.44±2.32	90.56±1.06
	IND : Banking split (75%) → OOD : Banking split (25%)								
Baseline	92.69±0.58	42.63±3.54	91.63±0.13	**41.32**±2.79	**92.48**±0.39	41.78±1.21	91.62±0.18	49.74±0.37	90.04±0.04
DOC (Shu et al., 2017)	91.83±0.21	51.05±3.54	90.29±1.34	**42.90**±5.58	**92.07**±0.93	51.06±3.91	89.97±1.27	49.87±5.40	90.48±1.16
ConSERT (Yan et al., 2021)	92.78±0.34	42.64±0.37	92.31±0.16	40.27±0.74	**92.73**±0.12	**39.41**±0.47	92.46±0.10	46.98±1.86	90.69±0.30
SimCSE (Gao et al., 2021)	92.70±0.33	43.55±1.30	92.39±0.02	**42.11**±1.12	**92.62**±0.05	43.23±0.28	92.29±0.18	47.17±2.14	90.36±0.18
MirrorBERT (Liu et al., 2021)	93.69±0.33	42.11±0.56	92.12±0.40	**40.00**±0.75	**92.33**±0.27	42.37±0.93	92.01±0.32	47.96±0.83	90.26±0.42
Li et al. (2021)	93.69±0.46	41.32±3.34	92.53±0.23	**40.92**±2.60	**92.63**±0.29	41.05±2.23	92.47±0.41	46.12±2.33	90.78±0.56
Zhou et al. (2021)	92.14±0.46	43.36±0.65	92.28±0.06	**41.58**±2.05	**92.42**±0.08	45.53±0.74	91.94±0.23	50.00±5.59	90.26±1.23
Zeng et al. (2021)	92.10±0.33	**48.16**±3.01	**88.04**±0.78	-	-	52.89±3.89	86.69±0.77	-	-
LaCL (ours)	94.55±0.33	**35.14**±2.79	**93.12**±0.04	38.42±1.48	91.84±0.19	41.84±1.30	92.40±0.75	60.79±0.75	85.73±0.10

Table 8: IND / OOD performance of each model 3 different settings on Far-OOD setting. The best performance in each method is indicated in **bold** and the global best is underlined. LaCL outperforms other methods constantly in both IND and OOD metric.

Error Type	Input Text	Ground Truth	Prediction
Keyword over reliance	Check **battery** health on this device	OOD	jump_start
	Read **text**	OOD	text
	Who invented the internet	OOD	who_made_you
	Where can i find cheap **rental** skis nearby	OOD	car_rental
	Search up someone who **plays** in a movie	OOD	play_music
	What **oil** is best for chicken	OOD	oil_change_how
	What is harry's real **name**	OOD	change_user_name
Mislabeled	Give me the weather forecast for today	OOD	weather
	I need you to order a new pair of eyeglasses for me	OOD	order
	Tell me something about linkin park	OOD	fun_fact
	What's my current electric bill	OOD	bill_balance
	Order me a book of stamps and envelopes	OOD	order

Table 9: Examples of OOD test samples misclassified as IND. The keywords that cause over-reliance are in **bold**.

Error Type	Input Text	Ground Truth	Prediction[†]
Misspelling	I **apprecaite** the help from you	thank_you	OOD / thank_you
	tell me how to **spent** frightened	spelling	OOD / spending_history
	I **appeciate** it	thank_you	OOD / yes
	How much is **alorie** intake	calories	OOD / calories
	Give me restaurant **reccomendations**	restaurant_suggestion	OOD / restaurant_suggestion
Nonstandard / Uncommon	**10-4**	yes	OOD / calculator
	Is it ok to use oil spray instead of **canola oil**	ingredient_substitution	OOD / oil_change_how
	What's your **bday**	how_old_are_you	OOD / what_are_your_hobbies
	This charge is **bs**	report_fraud	OOD / international_fees
	Ya	yes	OOD / goodbye
Absence of keywords	Tell fred that i don't have his guitar	text	OOD / find_phone
	Did i stick to my dinner budget	spending_history	OOD / spending_history
	Do i overspend when it comes to fast food	spending_history	OOD / spending_history
	I want to tell susan that the meeting has been cancelled	text	OOD / cancel_reservation
	That's all i need, i'm going now	goodbye	OOD / goodbye

[†] The right side of th indicates predicted label in IND before the input is sorted out as OOD by a threshold.

Table 10: Examples of IND test samples misclassified as OOD. Words related to their error type are highlighted in **bold**.