From Zero to Hero: Generalized Cold-Start Anomaly Detection

Anonymous ACL submission

Abstract

When first deploying an anomaly detection system, e.g., to detect out-of-scope queries in chatbots, there are no observed data, making data-driven approaches ineffective. Zero-shot anomaly detection methods offer a solution to such "cold-start" cases, but unfortunately they are often not accurate enough. This paper studies the realistic but underexplored *generalized cold-start* setting where an anomaly detection model is initialized using zero-shot guidance, but subsequently receives a small number of contaminated observations (namely, that may include anomalies). The goal is to make efficient use of both the zero-shot guidance and the observations. We propose ColdFusion, a method that effectively adapts the zero-shot anomaly detector to contaminated observations. To support future development of this new setting, we propose an evaluation suite consisting of evaluation protocols and metrics.

1 Introduction

Anomaly detection methods aim to flag data that violate accepted norms. For example, a customer support chatbot may be designed to answer queries about particular intents (in-scope) but not about other intents (out-of-scope). Unlike related tasks such as out-of-scope intent discovery and classification, which rely on large labeled in-scope data, anomaly detection approaches relax the labeling assumption and treat the problem as a one-class classification task (Lin et al., 2020; Zhang et al., 2021b; Mou et al., 2022; Zheng et al., 2020; Zhan et al., 2021; Lin and Xu, 2019; Zeng et al., 2021; Zhang et al., 2021a; Xu et al., 2020). Most anomaly detection methods (Reiss et al., 2021; Qiu et al., 2021; Zhang et al., 2023) require previous observations for training and are effective when many past observations are available. Such methods are not effective for systems just after deployment, as they lack access to any past observations. Zero-shot anomaly detection (Jeong et al., 2023; Li et al., 2024; Zhou et al., 2024) uses descriptions of the normal classes and does not require training data. While zero-shot methods can be used for freshly deployed systems, they result in reduced accuracy as the descriptions often fail to properly express the distribution of real data.

We explore the *generalized cold-start* setting which provides two types of guidance: i) a textual description of each normal class, serving as initial guidance, such as predefined topic names in chatbot systems; ii) a stream of t contaminated observations (that may include anomalies), e.g., real user queries. It is particularly relevant in real-world applications where, shortly after deployment, a short stream of user queries becomes available but the queries are not labeled into intent types and some of them are out-of-scope. To our knowledge, the only work that deals with a similar setting (Jeong et al., 2023) assumes prior knowledge of anomalies, that observations come from a single normal class and that they are not contaminated by anomalies.

To tackle the generalized cold-start setting, we present ColdFusion, a method for adapting a zero-shot model given the distribution of a limited observation stream. Our method is very effective, achieving considerably better results than pure zero-shot and observation-based methods. To encourage future research into this promising new setting, we provide evaluation protocols and metrics.

Our contributions are:

1. Proposing the new setting of generalized cold-start anomaly detection.
2. Presenting ColdFusion for tackling the setting.
3. Introducing a dedicated evaluation suite consisting of evaluation protocols and metrics.

2 Generalized Cold-Start Anomaly Detection

Task Definition. In the generalized cold-start setting, a model has access to K class descriptions

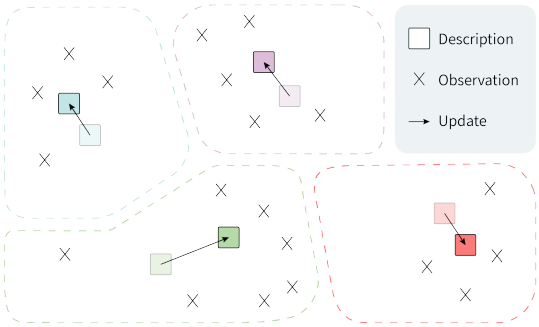


Figure 1: ColdFusion assigns each of the t observations to their nearest class, then adapts the embeddings of each class towards the assigned observations.

Algorithm 1: ColdFusion

Input: $\mathcal{D}_{\text{prior}}$, $\mathcal{D}_t$, p, query x.
Output: Anomaly score $S_{adapt}(x)$.
Step 1: Encode class descriptions and observations: $\phi(\mathcal{D}_{\text{prior}})$, $\phi(\mathcal{D}_t)$;
Step 2: Assign observations to classes based on nearest class descriptor: $a(x) = \arg\min_k \{d(\phi(x), \phi(c_k))\}_{k=1}^K$;
Step 3: Adapt class embeddings: $z_k = \text{median}(\phi(c_k), \{\phi(x) | a(x) = k\})$;
Step 4: Compute anomaly score for x: $S_{adapt}(x) = \min_k \{d(\phi(x), z_k)\}_{k=1}^K$;

$\mathcal{D}_{\text{prior}} = \{c_1, c_2, ..., c_K\}$ and a stream of t observations $\mathcal{D}_t = \{x_1, x_2, ..., x_t\}$, where t is small. We denote the percentage of anomalous observations as the *contamination ratio* $r\%$. An observation x either comes from one of the K normal classes or is anomalous, but we do *not* have access to the class or anomaly label. The task is to learn a model S to map each training sample x to an anomaly score such that high values indicate anomalous samples.

Application to chatbots. Our practical motivation is identifying out-of-scope queries in a recently deployed chatbot. We observe a stream of queries sent to the chatbot, as well as descriptions of all allowed intents. At time step $t+1$, we leverage both $\mathcal{D}_t$ and $\mathcal{D}_{\text{prior}}$ to classify a given query x_{t+1} as in-scope (INS) or out-of-scope (OOS).

3 Methodology

3.1 Recap: Zero-Shot Anomaly Detection

Zero-shot (ZS) anomaly detection maps each data point x to an anomaly score $S(x)$. Notably, ZS methods do not require past data, instead they are guided by a set of distinct normal class names $\{c_1, c_2, ..., c_K\}$ provided by the user. A pre-trained feature extractor ϕ maps each of the class names c_k, and observations x_t to deep embeddings $\phi(c_k)$ and $\phi(x_t)$. It then computes the distance d (often L_2 or Cosine) between the embeddings of the example and each of the class names. The final anomaly score is given by the distance to the nearest class:

$$S_{zs}(x) = \min_k \{d(\phi(x), \phi(c_k))\}_{k=1}^K \quad (1)$$

High anomaly scores serve as indicators of anomalies. The anomaly score can be converted to a binary label by choosing a threshold α such that $y = 0$ if $S(x) < \alpha$ and $y = 1$ if $S(x) \geq \alpha$.

Zero-shot anomaly detection can be used for OOS query detection by first specifying a set of allowed intents. Then a deep encoder extracts the embeddings of the target user query and intent descriptions. Finally, the method labels the user query as OOS if it is far from all allowed intent names.

3.2 Limitations of Existing Methods

In practice, it is impossible to provide perfect class descriptions, and therefore zero-shot anomaly detection often does not achieve sufficient accuracy. On the other hand, if the number of observations is limited, observation-based anomaly detection methods, such as K-nearest neighbors, struggle for three key reasons: i) the observations may not include all in-scope classes; ii) it is hard to estimate the true distribution of normal data from a few samples; iii) the observations may be contaminated, meaning they may include anomalies. Empirically, observation-based methods underperform ZS methods for small t (see Tab. 1 and Fig. 2).

3.3 Our Method: ColdFusion

To bridge the gap between ZS and observation-based methods, we propose ColdFusion (illustrated in Fig. 1), a method for generalized cold-start anomaly detection using domain adaptation. It improves ZS anomaly detection using the t observations in two key stages: i) assigning observations to classes; ii) adapting ZS class embeddings based on the assigned observations.

Assignment. We assign each of the t observations to the nearest class as measured in the feature space ϕ. We denote the class assignment of observation x as $a(x)$. More formally:

$$a(x) = \arg\min_k \{d(\phi(x), \phi(c_k))\}_{k=1}^K \quad (2)$$

We further define $\mathcal{C}_k$, the set of all observations assigned to class k as $\mathcal{C}_k = \{\phi(x) | a(x) = k\}$.

Encoder	Method	$AUC^2_{10\%}$			$AUC^2_{25\%}$			$AUC^2_{50\%}$			$AUC^2_{100\%}$		
		B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards
GTE	ZS	78.9	**83.1**	81.8	78.9	83.1	81.8	78.9	83.1	81.8	78.9	83.1	81.8
	DN2	76.7	64.8	70.0	76.2	76.0	75.6	75.9	80.2	79.6	75.3	82.2	80.2
	ColdFusion	**81.7**	82.3	**84.8**	**81.8**	**87.0**	**87.3**	**81.9**	**88.6**	**88.7**	**82.3**	**89.2**	**89.0**
MPNET	ZS	81.8	82.7	80.1	81.8	82.7	80.1	81.8	82.7	80.1	81.8	82.7	80.1
	DN2	78.3	69.7	69.8	78.2	78.9	76.9	77.6	82.3	80.9	76.3	83.6	81.1
	ColdFusion	**83.3**	**84.4**	**84.1**	**82.8**	**87.8**	**86.0**	**82.8**	**88.8**	**87.8**	**83.0**	**89.4**	**88.3**

Table 1: AUC^2_t results, with contamination of $r = 5\%$. Best results are in bold.

Method	$AUC^2_{10\%}$			$AUC^2_{25\%}$		
	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards
K-means	80.0	79.2	83.7	78.9	84.0	87.0
Mean	81.6	80.8	84.7	81.7	86.4	**87.5**
MI	81.6	**82.3**	**84.8**	**81.8**	**87.0**	87.1
Median	**81.7**	**82.3**	**84.8**	**81.8**	**87.0**	87.3

Table 2: AUC^2_t results, with contamination of $r = 5\%$ using the GTE model. MI refers to multiple iterations with median adaptation. Best results are in bold.

Adaptation. We now adapt each class embedding by considering both the initial class description and the assigned observations. Concretely, the adapted code for each class is the median of the set containing the embedding of the class descriptions and the embeddings of all assigned observations:

$$z_k = median(\{\phi(c_k)\} \cup \mathcal{C}_k) \quad (3)$$

We chose the median and not mean for contamination robustness. Note that this step will not modify the embedding of classes with no observations.

Anomaly scoring. ColdFusion uses the same anomaly scoring as ZS except that the class codes are the adapted $\{z_k\}_{k=1}^K$ instead of the encoding of the original description i.e., $S_{adapt}(x) = \min_k \{d(\phi(x_{t+1}), z_k)\}_{k=1}^K$.

4 Experiments

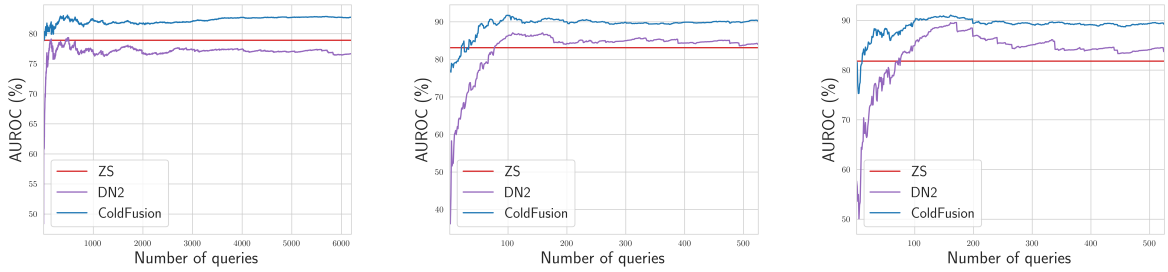
Experimental setting. Our experiments simulate the deployment of an OOS query detection system. We first randomly sort the queries so that each query has a unique time t, modeling a query stream. At each time t, we train a model using the t available observations and the K class names, and evaluate the model on the entire test set.

Datasets. We use three evaluation datasets, Banking77-OOS and CLINC-OOS segmented into CLINC-Banking and CLINC-Credit_Cards. Banking77-OOS (Casanueva et al., 2020; Zhang et al., 2021c) consists of $13,083$ customer service queries, categorized into 77 fine-grained intents within the online banking domain. Among these, 50 intents are in-scope, while the remaining 27 are OOS queries. CLINC-OOS (Larson et al., 2019; Zhang et al., 2021c), derived from the broader CLINC dataset, consists of two domains: "Banking" and "Credit cards", each featuring 10 in-scope and 5 OOS intents. The training sets for each domain include 500 in-scope queries, while the test sets contain 850 queries, with 350 designated as OOS instances. Notably, our setting is unsupervised i.e., observations do not include intent labels for training. Further details are in App. B.1.

Feature extractor & class encoding. We explored two feature encoders, namely the GTE model (Li et al., 2023) and MPNET (Song et al., 2020), both pre-trained on a large corpus of text pairs across various domains. We found that directly encoding intent topic names using these encoders did not meet our performance expectations (See Sec. 5). To overcome this challenge, we leverage ChatGPT to generate a query corresponding to each topic and utilize these generated queries as class descriptions instead of the intent topic names. For further details, please refer to App. A.

Baselines. We compare ColdFusion (Sec. 3.3) to several baselines. These include the zero-shot model (ZS), detailed in Sec. 3.1, which relies solely on the generated normal class descriptions. Additionally, we consider DN2, an observation-based anomaly detection method proposed by (Reiss et al., 2021). DN2 computes the anomaly score of an observation by its deep 1-nearest neighbor distance versus the previous observations $\mathcal{D}_t$. For implementation details refer to App. B.2.

Evaluation metrics. We propose a new metric to evaluate the cold-start setting, which emphasizes high-accuracy shortly after deployment (low t). At each time step t, we evaluate the performance of the anomaly scores model using the Area Under the Receiver Operation Characteristic (AUROC) curve.



(a) Banking77-OOS (b) CLINC-Banking (c) CLINC-Credit_Cards

Figure 2: Performance trends with contamination $r = 5\%$ using the GTE model over time demonstrate the superiority of our ColdFusion method over other baseline approaches.

We obtain an AUROC score for every time step, and we denote them as $\{\text{AUC}(t)\}_{t=1}^{T}$. We summarize this t vs. AUROC curve by the area under it up to time t. This is denoted as $\text{AUC}_{\tilde{t}}^2 = \frac{\sum_{t'=1}^{t} \text{AUC}(t')}{t}$, where $\tilde{t} = \frac{t}{T}$, the fraction of the training set used. The $\text{AUC}_{\tilde{t}}^2$ metric provides a concise summary of the model's accuracy freshly after deployment.

4.1 Results

We present our results in Tab. 1 and Fig. 2. ColdFusion consistently outperforms all baselines across the evaluated datasets by a large margin. Particularly, we see that DN2 performs poorly, especially with small t, and the zero-shot baseline (ZS) maintains constant performance over time. Conversely, our approach performs well even for low t values, and improves over time. The presence of anomalies in the data stream poses a challenge for DN2, as it solely relies on the observed contaminated stream. This reliance often leads to occasional decreases in performance for DN2, highlighting the vulnerability of methods that exclusively depend on the observed data without considering the underlying anomalies. Furthermore, our method's robustness to different feature encoders, as evidenced by consistent trends in both the GTE and MPNET models, suggests that it is not reliant on a single feature extractor. Results for different contamination are in App. C.

5 Ablation Study

Class embedding adaptation method. We investigate several variations of the adaptation method, shown in Tab. 2. i) Replacing our assignment and adaptation stages with K-means notably reduces performance, mainly due to its less effective random initialization method vs. our descriptor initialization; ii) Iterating multiple steps of assignment and adaptation, each time assigning to the adapted center, fails to outperform ColdFusion. The single iteration of ColdFusion is preferred, since multiple iterations increase the computational cost. Additionally, the results in Tab. 2 show that median adaptation is slightly better than using the mean on the evaluated datasets.

	Method	B77	C-Bank	C-Cards
GTE	Naive	76.9	60.7	69.8
	Generated	**78.9**	**83.1**	**81.8**
MPNET	Naive	79.8	69.6	73.7
	Generated	**81.8**	**82.7**	**80.1**

Table 3: Comparison of ZS models in terms of AUROC. As ZS models maintain constant performance over time and are not exposed to data, $\text{AUC}_{\tilde{t}}^2$ and contaminations are irrelevant. Best results are in bold.

Effectiveness of generated queries. In Tab. 3, we examine the impact of a naive ZS detector that simply encodes the intent names, compared to our ZS approach, which uses ChatGPT to generate a query for each intent and then encodes the generated query as the class embedding. The results highlight that naive encoding of intent names alone yields subpar performance, whereas our pre-processing procedure considerably improves results.

6 Conclusion

We introduced the new setting of generalized cold-start anomaly detection, modeling freshly deployed anomaly detection systems. Our proposed solution, ColdFusion, is a method for adapting zero-shot anomaly detection to align with an observation stream. We introduced an evaluation protocol and metrics for comparing future methods.

Limitations

Our proposed method has several limitations. i) Not all deployed anomaly detection systems encounter the generalized cold-start problem and indeed in the case where there are many observations and very few anomalies, it is sometimes better to use observation-driven methods e.g., DN2 (Reiss et al., 2021). However, we believe that it is a common issue, particularly in domains like chatbots; ii) Our approach relies on user-provided guidance for zero-shot detection, which may not be available in systems lacking such priors; iii) We assume a low contamination ratio; if this ratio is significantly higher, the effectiveness of our method may decrease.

Ethics Statement

Our work focuses on the development and evaluation of generalized cold-start anomaly detection methods, which have practical implications across various domains. Given the critical nature of anomaly detection systems, especially in applications where they may impact decision-making processes or user interactions, it is vital to consider ethical considerations at every stage of our research. Specifically, in the context of deploying anomaly detection systems, there is a risk of false positives or false negatives, which could lead to wrong outcomes. Therefore, it is crucial to thoroughly evaluate the robustness and reliability of our proposed methods, ensuring they perform effectively and equitably across diverse scenarios. From a societal perspective, our research contributes to the advancement of anomaly detection techniques, potentially enhancing the safety and security of systems deployed in various domains. Moreover, we recognize the importance of transparency in our research practices. We commit to openly sharing our findings, methods, and code to ensure reproducibility and enable further study by the research community.

References

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. 2002. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer.

Michael Glodek, Martin Schels, and Friedhelm Schwenker. 2013. Ensemble gaussian mixture models for probability density estimation. *Computational Statistics*, 28(1):127–138.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*.

Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616.

Ian Jolliffe. 2011. *Principal component analysis*. Springer.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. 2007. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer.

Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. 2024. Musc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *International Conference on Learning Representations*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. *arXiv preprint arXiv:1906.00434*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8360–8367.

Yutao Mou, Keqing He, Yanan Wu, Zhiyuan Zeng, Hong Xu, Huixing Jiang, Wei Wu, and Weiran Xu.

2022. Disentangled knowledge transfer for ood intent discovery with unified contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 46–53.

Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. 2021. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pages 8703–8714. PMLR.

Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. 2021. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814.

Tal Reiss and Yedid Hoshen. 2023. Mean-shifted contrastive loss for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2155–2162.

Lukas Ruff, Nico Gornitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*.

Bernhard Scholkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. 2000. Support vector method for novelty detection. In *NIPS*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning*.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1452–1460.

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. *arXiv preprint arXiv:2105.14289*.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. *arXiv preprint arXiv:2106.08616*.

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

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021b. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373.

Jianguo Zhang, Kazuma Hashimoto, Yao Wan, Zhiwei Liu, Ye Liu, Caiming Xiong, and Philip S Yu. 2021c. Are pretrained transformers robust in intent classification? a missing ingredient in evaluation of out-of-scope intent detection. *arXiv preprint arXiv:2106.04564*.

Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. 2023. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3914–3923.

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.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. 2024. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection.

A Zero-Shot Anomaly Detection

Zero-shot (ZS) anomaly detection assigns an anomaly score $S(x)$ to each data point x without relying on past data. Instead, it is guided by a set of class names $\{c_1, c_2, ..., c_K\}$ provided by the user. To tackle this challenge, we leverage ChatGPT to generate a user query corresponding to each class topic name. We use these generated queries as class descriptions instead of the intent topic names.

Query Generation: Utilizing ChatGPT-3.5, we generate a user query for each topic to serve as our class descriptions. Here, [DOMAIN] represents the chatbot domain (e.g., "Banking"). We employ the following template: "Generate queries that someone would ask a chatbot in [DOMAIN]. Generate one-sentence queries for each of the following topics: $\{c_1, c_2, ..., c_K\}$." This process yields a set of K user queries, denoted by $\{q_k\}_{k=1}^K$.

A pre-trained feature extractor ϕ maps each generated class name q_k and observation x to deep embeddings $\phi(q_k)$ and $\phi(x)$. Subsequently, we compute the L_2 distance between the example embeddings and each generated user query. The final anomaly score is determined by the distance to the nearest class:

$$S_{zs}(x) = \min_k \{d(\phi(x), \phi(c_k))\}_{k=1}^K$$

Alg. 2 outlines our zero-shot model.

A comparison between naive class names and generated queries is presented in Tab. 3.

Algorithm 2: Zero-Shot Detector

Input: $\mathcal{D}_{prior}$, ϕ, query x.
Output: Anomaly score $S_{zs}(x)$.
Step 1: Generate user queries using ChatGPT and $\mathcal{D}_{prior}$: $\{q_k\}_{k=1}^K$;
Step 2: Encode generated queries: $\{\phi(q_k)\}_{k=1}^K$ and input query: $\phi(x)$;
Step 3: Compute anomaly score for x: $S_{zs}(x) = \min_k \{d\phi(x), \phi(q_k))\}_{k=1}^K$;

B Experimental Details

B.1 Datasets

We employ three widely used datasets, Banking77-OOS and CLINC-OOS (which is split into CLINC-Banking and CLINC-Credit_Cards), to evaluate our anomaly detection approach.

Banking77-OOS. Banking77-OOS (Casanueva et al., 2020; Zhang et al., 2021c) is an annotated intent classification dataset designed for online banking queries. Comprising $13,083$ customer service queries, each query is labeled with one of 77 fine-grained intents within the banking domain. The dataset focuses on fine-grained, single-domain intent detection. Of these 77 intents, Banking77-OOS incorporates 50 in-scope intents, while the out-of-scope (OOS) queries are constructed based on 27 held-out in-scope intents. The training set consists of 5,095 in-scope user queries, and the test set comprises 3,080 user queries, including 1,080 OOS instances.

CLINC-OOS. CLINC-OOS (Larson et al., 2019; Zhang et al., 2021c) emanates from the broader CLINC dataset, encompassing 15 intent classes across 10 different domains, with integrated out-of-scope examples. For our evaluation, we focus on two domains: "Banking" and "Credit cards". Each domain is characterized by 5 in-scope and 10 out-of-scope intents. The training set for each domain comprises 500 in-scope user queries, while the test set includes 850 user queries, with 350 designated as out-of-scope instances.

B.2 Implementation Details & Baselines

Our implementation relies on two feature encoders: the GTE model (Li et al., 2023) and MPNET (Song et al., 2020), both pre-trained on a large corpus of text pairs across various domains. We use the `HuggingFace` library for both models. Specifically, for the GTE model, we employ the "thenlper/gte-large" model checkpoint, while for MPNET, we use the "sentence-transformers/all-mpnet-base-v2" model checkpoint. It's noteworthy that all baselines are using the same feature encoders in our comparisons. We use L_2 as a distance metric. For DN2 (Reiss et al., 2021), the implementation involves encoding $\mathcal{D}_t$ and the target query x with our feature encoder ϕ, followed by computing the 1-nearest-neighbor distance to $\phi(\mathcal{D}_t)$. We employ the `faiss` library for nearest-neighbor distance computations. In our ColdFusion in order to be robust to anomalies, we excluded observations assigned to class k but are further than τ. Formally, we define $\mathcal{C}_k$, as the set of all observations assigned to class k as:

$$\mathcal{C}_k = \{\phi(x) | a(x) = k, d(\phi(x), \phi(c_k)) \leq \tau\}$$

We set τ by first computing the distances between all samples and their assigned centers, sorting them, and choosing τ as the 90% percentile. An ablation study on this parameter is in App. C.

Encoder	Method	$AUC^2_{10\%}$			$AUC^2_{25\%}$			$AUC^2_{50\%}$			$AUC^2_{100\%}$		
		B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards
GTE	ZS	**78.9**	83.1	81.8	78.9	83.1	81.8	78.9	83.1	81.8	78.9	83.1	81.8
	DN2	74.6	71.2	71.6	77.5	79.4	79.1	78.8	82.9	82.9	79.2	84.7	85.7
	ColdFusion	79.0	**85.1**	**85.2**	**80.9**	**86.9**	**87.6**	**81.8**	**87.7**	**88.7**	**82.3**	**89.1**	**89.1**
MPNET	ZS	**81.8**	82.7	80.1	**81.8**	82.7	80.1	81.8	82.7	80.1	81.8	82.7	80.1
	DN2	76.6	74.7	70.7	79.1	82.4	78.5	80.1	85.7	82.7	80.5	86.9	84.8
	ColdFusion	80.6	**87.0**	**85.2**	81.7	**89.0**	**87.6**	**82.5**	**89.5**	**89.0**	**83.2**	**90.0**	**89.1**

Table 4: AUC^2_t results, with contamination of $r = 2.5\%$. Best results are in bold.

Encoder	Method	$AUC^2_{10\%}$			$AUC^2_{25\%}$			$AUC^2_{50\%}$			$AUC^2_{100\%}$		
		B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards
GTE	ZS	**78.9**	83.1	81.8	78.9	83.1	81.8	78.9	83.1	81.8	78.9	83.1	81.8
	DN2	70.6	67.2	71.3	72.5	77.5	78.1	73.4	80.5	81.1	73.8	80.8	81.9
	ColdFusion	77.4	**83.4**	**86.4**	**78.9**	**87.0**	**87.1**	**79.9**	**88.4**	**87.9**	**80.8**	**88.9**	**88.2**
MPNET	ZS	**81.8**	82.7	80.1	**81.8**	82.7	80.1	81.8	82.7	80.1	81.8	82.7	80.1
	DN2	72.3	72.8	70.7	74.5	82.4	78.0	75.3	84.8	80.9	75.3	84.1	80.9
	ColdFusion	79.9	**85.5**	**85.4**	81.1	**88.1**	**86.9**	**81.8**	**88.9**	**87.8**	**82.6**	**89.2**	**88.0**

Table 5: AUC^2_t results, with contamination of $r = 7.5\%$. Best results are in bold.

C More Results & Analysis

Contamination Ratios. We extend our analysis by considering additional contamination ratios of $r\% = 2.5$ and $r\% = 7.5$, as shown in Tables 4 and 5, respectively. Additionally, we present visual insights into ColdFusion's adaptive performance over time through the figures presented in Fig. 3, Fig. 4, Fig. 5, and Fig. 6. Across all contamination ratios, ColdFusion consistently outperforms all baselines by a significant margin, reinforcing our approach's robustness and effectiveness. These supplementary results further support the stability and reliability of ColdFusion's performance trends observed in the main analysis.

Effect of τ. Table 6 provides an ablation analysis of different τ parameters as defined in Eq. B.2. We observe that selecting the 50% and 75% percentiles yields suboptimal performance compared to using the 90% and 100% percentiles. These percentiles involve minimal filtering. Interestingly, there is a slight improvement in performance when employing the 90% percentile compared to the 100% percentile.

D Related Works

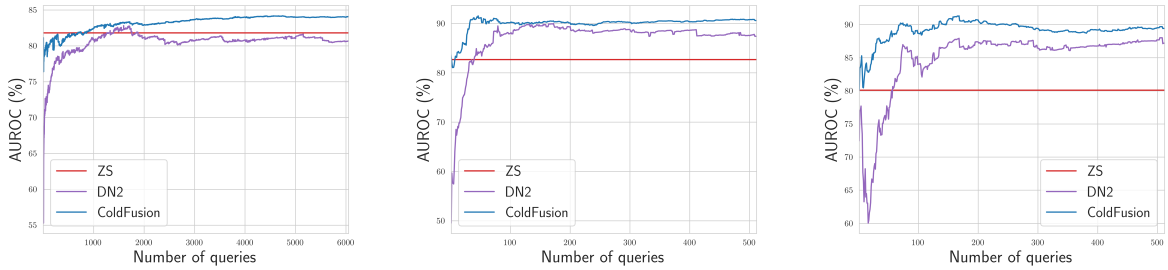
Out-of-scope intent discovery. Out-of-scope (OOS) intent discovery involves clustering new, unknown intents to identify potential development directions and expand the capabilities of dialogue systems. Prior works (Lin et al., 2020; Zhang et al., 2021b; Mou et al., 2022) in this domain have explored semi-supervised clustering using labeled in-domain data. Methods such as pre-training a BERT encoder with cross-entropy loss (Lin et al., 2020; Zhang et al., 2021b) and utilizing similarity constrained or supervised contrastive losses (Khosla et al., 2020) to learn discriminative features (Mou et al., 2022) aim to transfer intent representations. However, these approaches face challenges related to in-domain overfitting, where representations learned from in-scope data may not generalize well to OOS data. In contrast to this line of work, our approach focuses on detecting OOS intents rather than discovering them. Notably, our setting involves unlabeled in-scope intents, and our model's prior knowledge is limited to intent names.

Out-of-scope intent classification. OOS intent classification is categorized based on the use of extensive labeled OOS intent samples during training. The first category involves methods that use OOS samples during training, treating OOS intent classification as a $(n+1)$-class classification task (Zheng et al., 2020; Zhan et al., 2021). In contrast, the second category aims to minimize intra-class variance and maximize inter-class variance to widen the margin between in-scope and OOS intents (Lin and Xu, 2019; Zeng et al., 2021). Some approaches (Zhang et al., 2021a; Xu et al., 2020; Zeng et al., 2021) incorporate Gaussian distribution into the learned intent features to aid OOS detection. Our work stands apart from this line of research as it specifically addresses OOS intents, where in-scope

intents (topics) lack labels, and the model has no information or exposure to any OOS intents.

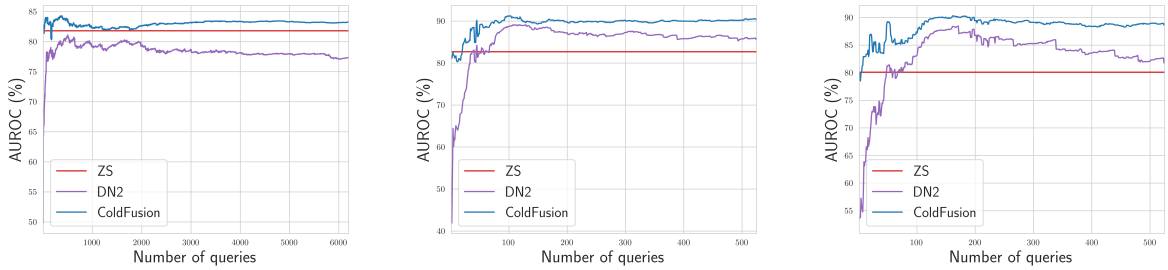
Classical anomaly detection methods. Detecting anomalies in images has been researched for several decades. The methods follow three main paradigms: i) Reconstruction - this paradigm first attempts to characterize the normal data by a set of basis functions and then attempts to reconstruct a new example using these basis functions, typically under some constraint such as sparsity or weights with a small norm. Samples with high reconstruction errors are atypical of normal data distribution and anomalous. Some notable methods include: principal component analysis (Jolliffe, 2011) and K-nearest neighbors (kNN) (Eskin et al., 2002); ii) Density estimation - another paradigm is to first estimate the density of normal data. A new test sample is denoted as anomalous if its estimated density is low. Parametric density estimation methods include Ensembles of Gaussian Mixture Models (EGMM) (Glodek et al., 2013), and non-parametric methods include kNN (which is also a reconstruction-based method) as well as kernel density estimation (Latecki et al., 2007). Both types of methods have weaknesses: parametric methods are sensitive to parametric assumptions about the nature of the data whereas non-parametric methods suffer from the difficulty of accurately estimating density in high-dimensions; iii) One-class classification (OCC) - this paradigm attempts to fit a parametric classifier to distinguish between normal training data and all other data. The classifier is then used to classify new samples as normal or anomalous. Such methods include one-class support vector machine (OCSVM) (Scholkopf et al., 2000) and support vector data description (SVDD) (Tax and Duin, 2004).

Deep learning for anomaly detection. This line of work is based on the idea of initializing a neural network with pre-trained weights and then obtaining stronger performance by further adaptation of the training data. DeepSVDD (Ruff et al., 2018) suggested to first train an auto-encoder on the normal training data, and then using the encoder as the initial feature extractor. Moreover, since the encoder features are not specifically fitted to anomaly detection, DeepSVDD adapts to the encoder training data. However, this naive training procedure leads to catastrophic collapse. An alternative direction is to use features learned from auxiliary tasks on large-scale external datasets. Transferring pre-trained features for out-of-distribution detection has been proposed by (Hendrycks et al., 2019). It was recently established (Reiss et al., 2021) that given sufficiently powerful representations, a simple criterion based on the kNN distance to the normal training data achieves strong performance. The best performing methods (Reiss et al., 2021; Reiss and Hoshen, 2023) combine pre-training on external datasets and a second finetuning stage on the provided normal samples in the training set, but they require many data observations and assume that the observations are not contaminated.



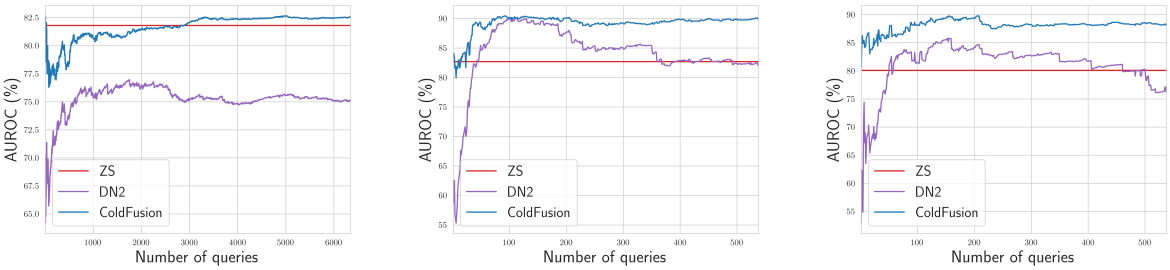
(a) Banking77-OOS (b) CLINC-Banking (c) CLINC-Credit_Cards

Figure 3: Performance trends with contamination $r = 2.5\%$ using the MPNET model over time.



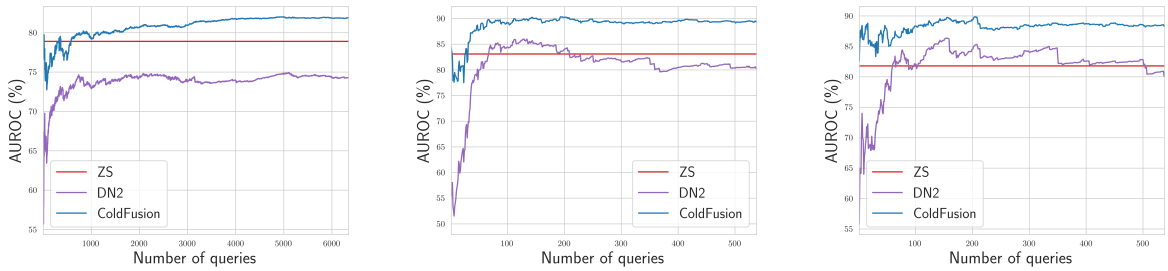
(a) Banking77-OOS (b) CLINC-Banking (c) CLINC-Credit_Cards

Figure 4: Performance trends with contamination $r = 5\%$ using the MPNET model over time.



(a) Banking77-OOS (b) CLINC-Banking (c) CLINC-Credit_Cards

Figure 5: Performance trends with contamination $r = 7.5\%$ using the MPNET model over time.



(a) Banking77-OOS (b) CLINC-Banking (c) CLINC-Credit_Cards

Figure 6: Performance trends with contamination $r = 7.5\%$ using the GTE model over time.

τ	$\text{AUC}^2_{10\%}$			$\text{AUC}^2_{25\%}$			$\text{AUC}^2_{50\%}$			$\text{AUC}^2_{100\%}$		
	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards	B77	C-Bank	C-Cards
$\tau = perc(\phi(\mathcal{D}_t), 50\%)$	80.1	80.4	80.7	80.6	83.6	83.0	80.7	85.1	84.6	81.3	86.5	85.4
$\tau = perc(\phi(\mathcal{D}_t), 75\%)$	81.8	81.9	83.0	81.9	85.5	84.8	81.8	87.4	86.7	82.1	88.2	87.7
$\tau = perc(\phi(\mathcal{D}_t), 100\%)$	**82.0**	81.1	**85.0**	**82.1**	86.0	86.7	81.8	88.0	88.0	**82.3**	89.0	88.5
$\tau = perc(\phi(\mathcal{D}_t), 90\%)$	81.7	**82.3**	84.8	81.8	**87.0**	**87.3**	**81.9**	**88.6**	**88.7**	**82.3**	**89.2**	**89.0**

Table 6: $\text{AUC}^2_{\tilde{t}}$ results using the GTE model, with contamination of $r = 5\%$. Best results are in bold.