Beyond Outlier Detection: Outlier Interpretation by Attention-Guided Triplet Deviation Network

Hongzuo Xu
Science and Technology on Parallel and Distributed Processing Laboratory
College of Computer, National University of Defense Technology
Changsha, China
xuhongzuo13@nudt.edu.cn

Yijie Wang*
Science and Technology on Parallel and Distributed Processing Laboratory
College of Computer, National University of Defense Technology
Changsha, China
wangyijie@nudt.edu.cn

Songlei Jian*
College of Computer, National University of Defense Technology
Changsha, China
jiansonglei@nudt.edu.cn

Zhenyu Huang
College of Computer, National University of Defense Technology
Changsha, China
huangzhenyu15@foxmail.com

Yongjun Wang
College of Computer, National University of Defense Technology
Changsha, China
wangyongjun@nudt.edu.cn

Ning Liu
College of Computer, National University of Defense Technology
Changsha, China
liuning17a@nudt.edu.cn

Fei Li
Alibaba Cloud Computing Co. Ltd.
Beijing, China
renlei.lf@alibaba-inc.com

ABSTRACT

Outlier detection is an important task in many domains and is intensively studied in the past decade. Further, how to explain outliers, i.e., *outlier interpretation*, is more significant, which can provide valuable insights for analysts to better understand, solve, and prevent these detected outliers. However, only limited studies consider this problem. Most of the existing methods are based on the score-and-search manner. They select a feature subspace as interpretation per queried outlier by estimating outlying scores of the outlier in searched subspaces. Due to the tremendous searching space, they have to utilize pruning strategies and set a maximum subspace length, often resulting in suboptimal interpretation results. Accordingly, this paper proposes a novel Attention-guided Triplet deviation network for Outlier interpretatioN (ATON). Instead of searching a subspace, ATON directly learns an embedding space and learns how to attach attention to each embedding dimension (i.e., capturing the contribution of each dimension to the outlierness of the queried outlier). Specifically, ATON consists of a feature embedding module and a customized self-attention learning module, which are optimized by a triplet deviation-based loss function. We obtain an optimal attention-guided embedding space with expanded high-level information and rich semantics, and thus outlying behaviors of the queried outlier can be better unfolded. ATON finally distills a subspace of original features from the embedding module and the attention coefficient. With the good generality, ATON can be employed as an additional step of any black-box outlier detector. A comprehensive suite of experiments is conducted to evaluate the effectiveness and efficiency of ATON. The proposed ATON significantly outperforms state-of-the-art competitors on 12 real-world datasets and obtains good scalability w.r.t. both data dimensionality and data size.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Anomaly detection**.

KEYWORDS

Outlier interpretation, Feature embedding, Self-attention, Triplet deviation

ACM Reference Format:
Hongzuo Xu, Yijie Wang, Songlei Jian, Zhenyu Huang, Yongjun Wang, Ning Liu, and Fei Li. 2021. Beyond Outlier Detection: Outlier Interpretation by Attention-Guided Triplet Deviation Network. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449868

1 INTRODUCTION

Outlier detection is an important research field in data science, which identifies uncommon data objects that deviate significantly from the majority [23]. Data is often generated to reflect activities in the system or observations of the entities [1], and the appearance of outliers indicates unusual data generating process or even severe faults and potential threats (e.g., abnormal web traffic might be network attacks, illegal operations in the stock market can cause

[*] Corresponding authors: Yijie Wang and Songlei Jian.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449868

serious economic damage, and a glitch in industrial control systems may trigger big disasters). Outlier detection algorithms are successfully applied to these real-world scenarios, preventing various kinds of risks without expensive human surveillance.

However, outlier detection is a half-done problem. The interpretation of detected outliers is an essential complementary task. It is hard for analysts to understand why a data object has been considered to be an outlier by solely utilizing detection results (predict label or outlier score) but without any clues. To perceive one outlier, analysts need to examine its behavior in every individual feature or even every possible feature combination, which is a laborious work, especially in high-dimensional data. Hence, it is more important to investigate what distinguishes the queried outlier from the given dataset and how to characterize the queried outlier, namely *outlier interpretation*. This task is also referred to as outlier explanation [13], outlier aspect mining/discovering [6, 28], outlier property detection [2], and outlier description [14]. Given interpretations of detected outliers, analysts can better understand these outliers and further choose to trust or ignore them. Downstream troubleshooting and decision-making will be more efficient. Mechanisms (systems, regulations, or policies) can also be optimized to prevent such outliers better. This task has broad real-world application in failure diagnosis of cloud service systems. Further, JointCloud [32] is an emerging architecture that empowers the cooperation among multiple clouds to provide efficient cross-cloud services. This architecture naturally contains complicated topology and service-calling dependencies. Outlier detection and interpretation form an important "regulatory" module in JointCloud architecture to ensure the reliability of cross-cloud services. Intelligent failure diagnosis technologies are urgently required in JointCloud systems to automatically provide analysts with deeper and clearer insight of outliers.

Following [17, 20], this paper also formally defines an interpretation of the queried outlier as a tailored feature subspace where the outlierness of this outlier is well exhibited. We can also cast outlier interpretation as a problem of computing the contribution of each feature to the outlierness of the queried outlier. Feature subspace can be further obtained by incorporating a threshold setting approach. Therefore, the ground-truth interpretation of the queried outlier can be obtained by selecting the best subspace from the power set of the original feature space. The queried outlier can be easily identified in this subspace by human analysts or outlier detectors. As shown in Figure 1 (a), the queried outlier is initially described by three features f_1, f_2, and f_3, and the interpretation is feature subspace $\{f_1, f_2\}$.

Although we have had a long list of various kinds of outlier detectors, comparably sparse literature considers the problem of explaining outliers detected by any outlier detector. The mainstream of prior art is based on the score-and-search manner [6, 12, 28, 31], which generally only leads to suboptimal results. These methods search possible feature subspaces and compute the outlying degree of the queried outlier in each subspace. However, they have to employ searching strategies, impose pruning methods, and set a maximum subspace length to handle tremendous searching space (the number of possible subspaces increases exponentially with the growth of data dimensionality). Therefore, they may fail to obtain the optimal interpretation subspace. As shown in Figure 1 (b), f_1 and f_3 are pruned in the first level, and thus this interpretation method can only yield a suboptimal feature subspace $\{f_1, f_2\}$, failing to retrieve the best subspace $\{f_1, f_3\}$.

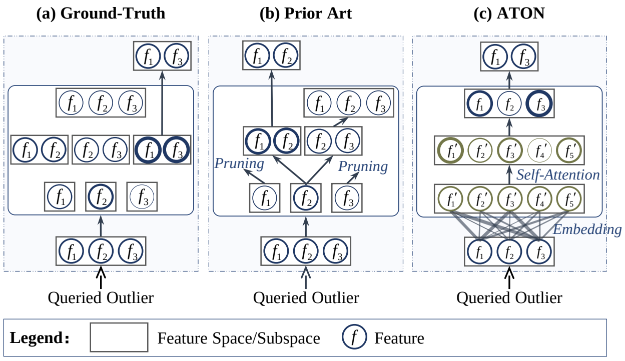

Figure 1: Ground-Truth Interpretation and Comparison Between Prior Art and the Proposed ATON. Line width of each feature represents the contribution to the outlierness of the queried outlier. (a) Ground-truth interpretation is obtained by selecting the best subspace from the power set of the feature space. (b) Prior art is based on subspace searching, which may neglect the accurate interpretation subspace because of applied pruning methods. (c) The proposed ATON learns an optimal attention-guided embedding space.

In light of this limitation, this paper proposes a novel Attention-guided Triplet deviation network for Outlier interpretatioN (ATON for short). Compared with conventional subspace searching-based approaches, ATON directly learns an embedding space and learns how to attach attention to new dimensions. We illustrate the basic insight of ATON in Figure 1 (c). ATON first constructs a feature embedding module to convert original feature space ($\{f_1, f_2, f_3\}$) to a new embedding space ($\{f'_1, f'_2, f'_3, f'_4, f'_5\}$). Note that these new dimensions can be embedded with expanded high-level feature patterns and rich semantics, and thus it is easier to capture and analyze outlying behaviors of the queried outlier in this embedding space. We then propose a customized self-attention learning module to attach attention to each new embedding dimension (feature f'_1, f'_3, and f'_5 are with higher attention coefficient). This module learns the contribution of each dimension to the outlierness of the queried outlier. In ATON, the queried outlier is combined with heuristically sampled informative normal data to generate a group of triplets. We propose a triplet deviation-based loss function, which estimates the separability of the queried outlier and its normal counterparts within the triplets. The feature embedding module and the self-attention module can then be constantly optimized to find an optimal embedding space with attached attention. It is noteworthy that ATON has a continuous solution space. By contrast, the solution space is generally discretized (each feature is either retained or removed) in prior art. Arguably, it is a superiority because fine-grained optimization procedures can generally produce better results. Feature weights of original space can be finally distilled from the embedding module and the learned feature attention coefficient (original feature f_1 and f_3 have higher weight). Interpretation subspace $\{f_1, f_3\}$ can then be derived by incorporating a proposed threshold setting approach.

Overall, our main contributions are summarized as follows:

- We propose an outlier interpretation method ATON having good generality. ATON is model-agnostic, i.e., it can be employed as an additional step to explain outliers detected by any black-box outlier detection algorithm. ATON is also domain-agnostic, i.e., it suits for data from various domains.
- ATON transforms the original feature space to an embedding space with expanded high-level information and rich semantics by harnessing the representation power of neural networks. In this converted embedding space, the contribution of each dimension to the outlierness of the queried outlier is automatically learned by the customized self-attention learning module.
- ATON obtains good scalability since it avoids time-consuming subspace searching process. Besides, ATON generates triplets by only sampling limited informative normal data. As a result, ATON has linear time complexity w.r.t. data size (in datasets with fixed ratios of queried outliers).
- ATON is evaluated by a comprehensive suite of experiments with good reproducibility[1]. To the best of our knowledge, it is the first work that releases the ground-truth outlier interpretation annotations of real-world datasets, which fosters further research on this practical problem.

Extensive experiments show that ATON achieves impressive performance leap over the state-of-the-art competitors. We also use case studies to visually demonstrate interpretation quality. Ablation study validates the significance of each key design of ATON. We then investigate the effect of hyper-parameters and give recommended settings. ATON obtains good scalability w.r.t. both data dimensionality and data size in scale-up test.

2 RELATED WORK

Outlier detection is intensively studied by the community in the past decade. There are various kinds of outlier detection algorithms, e.g., probability-based methods [15], ensemble-based methods [16, 35], and deep learning-based methods [22, 24]. Note that some detectors, e.g., [4, 5], also consider the interpretability of the detected outliers, and some categorical-data-oriented outlier detectors [10, 33, 34] have intrinsic interpretable results. However, they cannot explain the detection results produced by any black-box detectors.

By contrast, limited studies consider model-agnostic outlier interpretation. Prior art generally utilizes score-and-search manner. The methods in this research line search a subspace where the queried outlier obtains the highest outlier score. They propose new efficient outlying scoring functions combining with some searching strategies. For example, Duan et al. [6] introduce OAMiner that employs kernel density estimation as scoring function and uses a pruning method based on anti-monotonicity properties. The work of [31] proposes two outlying scoring functions (density Z-score and isolation path length) and employs beam search. Keller et al. [12] use the adaptive subspace searching based on random subspace sampling. A very recent method, SiNNE [28], computes isolation score using the nearest neighbor ensemble and also uses beam search. Kuo and Davidson [14] introduce a constraint programming approach, which can also be deemed as a subspace searching process. These subspace searching-based methods might fail to obtain an accurate interpretation subspace because of the applied searching strategies and pruning methods. They also need to restrict a maximum subspace length in practical usage.

Some methods attempt to obtain more efficient interpretation by employing feature selection techniques and sparse classifiers. COIN [17] maps the interpretation task to a classification problem. It trains a set of ℓ_1-norm classifiers that separate augmented outlier data from clusters of nearby normal data. The linear weights in classifiers are used to explain the queried outlier. The work in [20] employs feature selection techniques and classifiers to separate the queried outlier and its surroundings. However, these two methods might still fail to obtain sufficient performance because: (i) They augment the queried outlier to form a hypothetical class so that the classifier can work properly, which means the quality of data augmentation largely determines the effectiveness of interpretation; and (ii) A powerful classifier can still yield good classification results even if in a low-quality feature subspace.

There are some approaches producing some new forms of explanation, e.g., focus plot (2-d scatter plot) in [9], packs (hyper-ellipsoid in a feature subspace) in [19], and feature sequences in [30]. Besides, decision tree has intrinsic explanation ability, which is employed to extract explanation rules in [13, 25]. There are also some model-specific or domain-specific outlier interpretation methods. The method in [11] is tailored for one-class SVMs, which "neuralizes" the predictions and uses deep Taylor decomposition to obtain explanations. The outliers detected by GRU-based autoencoder are explained in [7]. An outlier contribution explainer is proposed in [36] especially for cyber-security applications.

Interpretable machine learning is a related field, which studies possible explanations for the predictions generated by machine learning algorithms. LIME [27] is a well-known explainer for arbitrary classifier prediction. LIME learns a local interpretable model around the prediction to infer the explanations. A more advanced approach SHAP is proposed in [18]. SHAP is a game theoretic approach to learn the optimal Shapley values (the contribution of each feature) as explanations.

3 PROBLEM STATEMENT

Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ be an input dataset containing a set of data objects, where $|\mathcal{X}| = N$. Each data object $\boldsymbol{x} \in \mathcal{X}$ is a real-valued feature vector described by D features, i.e., $\boldsymbol{x} \in \mathbb{R}^D$. The feature set is denoted as $\mathcal{F} = \{f_1, f_2, \cdots, f_D\}$. The data objects in a small subset of dataset $\mathcal{X}_o \subset \mathcal{X}$ are outliers, where $|\mathcal{X}_o| = N_o$ and $N_o \ll N$.

Outlier interpretation is to explain why a data object is an outlier. Inspired by the taxonomy of interpretable machine learning [21], this problem can also be divided into model-agnostic and model-specific interpretation based on whether the queried outlier is detected by a specific outlier detection model. This paper aims to address the model-agnostic outlier interpretation problem because the approaches in this category have better generality and can be incorporated with any existing outlier detector.

Following [17, 20, 28], We formally define the outlier interpretation task as follows:

[1] Our source code is available at https://github.com/xuhongzuo/outlier-interpretation.

Definition 3.1 (Outlier interpretation). *Given a dataset $\mathcal{X}$ and a queried outlier set $\mathcal{X}_o \subset \mathcal{X}$, outlier interpretation OI is to find a tailored explanatory subspace $\mathcal{E} \subseteq \mathcal{F}$ for each outlier in query set $\boldsymbol{o} \in \mathcal{X}_o$. The outlierness of the queried outlier $\boldsymbol{o}$ can be explicitly unfolded in this feature subspace $\mathcal{E}$ such that human analysts or outlier detection algorithms OD can easily and accurately predict the queried outlier. This procedure is formally represented as:*

$$\mathcal{E} = OI(\boldsymbol{o}|\mathcal{X}), \boldsymbol{o} \in \mathcal{X}_o \tag{1}$$

s.t.

$$OD(\boldsymbol{o}_{\mathcal{E}}|\mathcal{X}_{\mathcal{E}}) = outlier \tag{2}$$

where $\mathcal{X}_{\mathcal{E}}$ and $\boldsymbol{o}_{\mathcal{E}}$ denote data object(s) described in the interpretation feature subspace $\mathcal{E}$.

Generally, features of real-world tabular datasets have real semantics. For example, in the field of AI for IT operations (AIOps), the data records in distributed tracing systems are response times of different services. Also, in the medical diagnosis area, the features in the Breast Cancer dataset are computed from a digitized image of a breast mass to describe characteristics of the cell nuclei. Therefore, given the behavior (feature value) in selected features, the human analysts can locate the root cause of an outlier, thereby explain this outlier.

Interpretation methods can also output feature weight as an explanation apart from directly producing explanatory subspace. The feature weight is seen as the contribution of each feature to the outlierness. These methods can further yield explainable feature subspace when combining with a threshold setting approach. It is similar to the problem setting of outlier detection, i.e., some outlier detectors output outlier scores for data objects while the other detectors directly differentiate whether a data object is an outlier.

4 OUTLIER INTERPRETATION NETWORK

In this section, we first introduce the network architecture of ATON, then present the specific methodology, and finally give the pseudo code of our algorithm.

4.1 Network Architecture

Based on the problem definition of outlier interpretation, we aim to learn a feature subspace to interpret each queried outlier, where the target outlier can be easily separated from normal data. We introduce an Attention-guided Triplet deviation network for Outlier interpretatioN (ATON for short). ATON uses the proposed triplet deviation-based loss to optimize the feature embedding module and the customized self-attention learning module. In embedding space with attached attention, ATON attempts to differentiate the queried outlier and its normal counterparts of each generated triplets. The interpretable feature subspace of this queried outlier is finally distilled from the optimized feature embedding module and the learned self-attention coefficient. The network structure of ATON is shown in Figure 2, which consists of four main components:

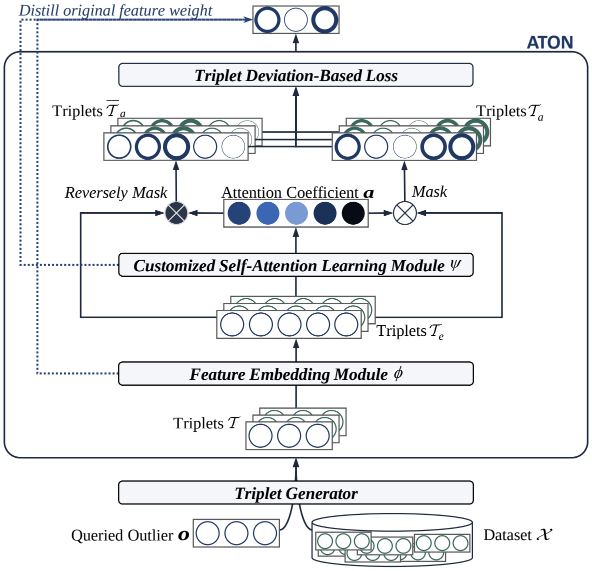
Distill original feature weight
ATON
Triplet Deviation-Based Loss
Triplets $\bar{\mathcal{T}}_a$
Triplets $\mathcal{T}_a$
Reversely Mask
Attention Coefficient $\boldsymbol{a}$
Mask
Customized Self-Attention Learning Module ψ
Triplets $\mathcal{T}_e$
Feature Embedding Module ϕ
Triplets $\mathcal{T}$
Triplet Generator
Queried Outlier $\boldsymbol{o}$
Dataset $\mathcal{X}$

Figure 2: Network Architecture of ATON

- Given a queried outlier $\boldsymbol{o}$ and the dataset $\mathcal{X}$, we first select normal data objects from dataset $\mathcal{X}$ to generate a set of triplets $\mathcal{T}$ as training data, i.e., $\mathcal{T} = \{\langle \boldsymbol{o}, \boldsymbol{m}, \boldsymbol{n}\rangle\}, \boldsymbol{m}, \boldsymbol{n} \in \mathcal{X}$.
- Subsequently, the data objects in the generated triplets are converted to a new embedding space by the feature embedding module $\phi : \mathbb{R}^D \to \mathbb{R}^d$. We get a set of embedded triplets $\mathcal{T}_e = \{\langle \phi(\boldsymbol{o}), \phi(\boldsymbol{m}), \phi(\boldsymbol{n})\rangle\}$. High-level information and rich semantics are expected to be expanded and directly demonstrated in this embedding space.
- ATON constructs a customized self-attention learning module $\psi : \langle \mathbb{R}^d, \mathbb{R}^d, \mathbb{R}^d \rangle \to \mathbb{R}^d$ afterward. Attention coefficient vector $\boldsymbol{a} \in \mathbb{R}^d$ is derived from the embedded triplets $\boldsymbol{a} = \psi(\mathcal{T}_e)$. $\boldsymbol{a}$ is a real-valued vector that measures the contribution of each embedding dimension to the outlierness of the queried outlier. ATON masks each element in triplets using attention coefficient vector $\boldsymbol{a}$ and outputs a set of attention-guided triplets $\mathcal{T}_a = \left\{\langle \boldsymbol{o}^{\text{attn}}, \boldsymbol{m}^{\text{attn}}, \boldsymbol{n}^{\text{attn}}\rangle\right\}$. We also obtain another set of triplets $\bar{\mathcal{T}_a} = \left\{\langle \boldsymbol{o}^{\text{r-attn}}, \boldsymbol{m}^{\text{r-attn}}, \boldsymbol{n}^{\text{r-attn}}\rangle\right\}$ that is reversely masked by the attention coefficient.
- ATON then calculates triplet deviation-based loss to assess triplet separability in masked data $\mathcal{T}_a$ and reversely masked data $\bar{\mathcal{T}_a}$. Parameters in previous steps are constantly optimized using this loss function.

4.2 Specific Model Design

In this section, we introduce the specific model design of ATON. Four main components of the network are presented in turn.

4.2.1 Triplet Generator. A set of triplets $\mathcal{T}$ is first generated as training data. These triplets are utilized to learn the separability of the queried outlier from the normality. Thus, the first triplet position is fixed as the queried outlier $\boldsymbol{o}$, and the rest two positions should well represent the normal data. We simultaneously consider two factors, namely general normality of the dataset and the local normality of the queried outlier. Two candidate sets are formed for the rest two positions. $\mathcal{X}_{\text{random}}$ denotes a candidate set of normal data objects randomly sampled from the full dataset, representing the general normality. In terms of the local normality, the nearest neighbor normal data objects of the queried outlier $\boldsymbol{o}$ are gathered in a candidate set $\mathcal{X}_{\text{neighbor}}$. We use Euclidean distance to define the distance of two data objects when searching the nearest neighbors. The set of generated triplets $\mathcal{T}$ is denoted as follows:

$$\mathcal{T} = \left\{ \gamma \middle| \gamma = \langle \boldsymbol{o}, \boldsymbol{m}, \boldsymbol{n} \rangle, \forall \boldsymbol{m} \in \mathcal{X}_{\text{random}}, \forall \boldsymbol{n} \in \mathcal{X}_{\text{neighbor}} \right\}. \tag{3}$$

For simplicity, two candidate sets ($\mathcal{X}_{\text{random}}$ and $\mathcal{X}_{\text{neighbor}}$) are restricted to be with the same cardinality. We use r to denote this sampling size, which is a parameter of ATON. Thus, the size of generated triplet set $\mathcal{T}$ is r^2. Abundant training data normally brings better performance, but too large r also results in low efficiency. We conduct a parameter test in Section 5.5 to investigate the effect of this parameter empirically.

4.2.2 Feature Embedding Module. Feature embedding is to convert the original features to a new feature space by a mapping function ϕ. ATON further attaches attention to each embedding dimension. Thus, We need to transform the attention coefficients of these embedding dimensions back to the original features by following some rules. This requirement should be considered when specifying this component.

ATON uses one learnable linear layer as feature mapping function ϕ, parameterized by a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times D}$, to obtain a new embedding space with powerful representation ability. Simple linear transformation can also ensure the scalability of ATON.

Data object $\boldsymbol{x}$ is transferred to a new embedding space via feature mapping function ϕ as:

$$\phi(\boldsymbol{x}) = \begin{bmatrix} \mathbf{W}(1,1)\boldsymbol{x}(1) + \mathbf{W}(1,2)\boldsymbol{x}(2) + \cdots + \mathbf{W}(1,D)\boldsymbol{x}(D) \\ \mathbf{W}(2,1)\boldsymbol{x}(1) + \mathbf{W}(2,2)\boldsymbol{x}(2) + \cdots + \mathbf{W}(2,D)\boldsymbol{x}(D) \\ \cdots \\ \mathbf{W}(d,1)\boldsymbol{x}(1) + \mathbf{W}(d,2)\boldsymbol{x}(2) + \cdots + \mathbf{W}(d,D)\boldsymbol{x}(D) \end{bmatrix} \tag{4}$$

where $\mathbf{W}(i,j)$ indicates the element in the i-th row and the j-th column of matrix $\mathbf{W}$, and $\boldsymbol{x}(i)$ denotes the i-th element of vector $\boldsymbol{x}$. Each dimension can be seen as a linear pattern (combination) of the original feature space.

All the data objects in generated triplets are embedded into this new feature space to derive an embedded triplet set $\mathcal{T}_e$ as:

$$\mathcal{T}_e = \left\{ \gamma' \middle| \gamma' = \langle \phi(\boldsymbol{o}), \phi(\boldsymbol{m}_i), \phi(\boldsymbol{n}_j) \rangle, i, j \in \{1, 2, \cdots, r\} \right\}. \tag{5}$$

The dimensionality of the embedding space d is a parameter of ATON. The network can generate plentiful linear combinations of original features when giving a larger d. An empirical study of parameter d is also included in Section 5.5.

4.2.3 Customized Self-Attention Learning Module. A customized self-attention learning module ψ is constructed in this step, which aims to measure the contribution of each new embedding dimension to the outlierness of the queried outlier. Attention mechanism has been a *de facto* standard in many sequence-based tasks, e.g., language modeling and machine translation, to focus on the most relevant part of the input to make decisions [3].

We construct a single-hidden-layer feedforward neural network as the customized self-attention learning module ψ in ATON. Hidden layer f_1 is with h_1 hidden units, which is parameterized by weight matrix $\mathbf{W}_a \in \mathbb{R}^{h_1 \times 3d}$ and ReLU activation function σ. We directly set $h_1 = \lfloor 1.5d \rfloor$ for simplicity. The weight matrix of output layer f_2 is denoted as $\mathbf{W}'_a \in \mathbb{R}^{d \times h_1}$. This self-attention learning module finally uses a column-wise min-max normalization function δ to scale the attention coefficient to $[0, 1]$.

Before entering the self-attention module, all the triplets are first flattened to vectors. Triplet $\langle \phi(\boldsymbol{o}), \phi(\boldsymbol{m}_i), \phi(\boldsymbol{n}_j) \rangle$ in $\mathcal{T}_e$ is converted to an individual vector $\boldsymbol{u} \in \mathbb{R}^{3d}$ as:

$$\boldsymbol{u} = [\phi(\boldsymbol{o}) \| \phi(\boldsymbol{m}_i) \| \phi(\boldsymbol{n}_j)] \tag{6}$$

where $[\cdot \| \cdot \| \cdot]$ represents vector concatenation. We obtain a set of flattened vectors with size $|\mathcal{T}_e| = r^2$. A matrix $\mathbf{U} \in \mathbb{R}^{3d \times r^2}$ is used to contain all the vectors derived from the triplet set.

Fully expanded out, the network first produces an attention matrix $\mathbf{A} \in \mathbb{R}^{d \times r^2}$ as follows:

$$\begin{aligned} \mathbf{A} = \psi(\mathcal{T}_e) &= (f_1 \circ f_2)(\mathcal{T}_e) \\ &= \delta\Big(\mathbf{W}'_a\big(\sigma(\mathbf{W}_a \mathbf{U})\big)\Big). \end{aligned} \tag{7}$$

The attention coefficient vector $\boldsymbol{a} \in \mathbb{R}^d$ is the average of the columns of matrix $\mathbf{A}$:

$$\boldsymbol{a} = \frac{1}{r^2} \sum_{i=1}^{r^2} \mathbf{A}(\cdot, i) \tag{8}$$

where $\mathbf{A}(\cdot, i)$ is the i-th column of matrix $\mathbf{A}$ indicating the attention coefficient of one triplet.

Attention weight is directly derived from the triplets. Hierarchical complex interactions and relationships of intra-object and inter-object dimensions are modeled during the attention learning process thanks to the connectivity property of neural network.

After getting attention coefficient vector $\boldsymbol{a}$, the data objects in triplets can be masked to create an attention-guided space. Note that we also use attention coefficient vector $\boldsymbol{a}$ to reversely mask the triplets. ATON yields attention-guided triplet set $\mathcal{T}_a$ and reverse-attention-guided triplet set $\bar{\mathcal{T}_a}$ as follows.

$$\begin{aligned} \mathcal{T}_a &= \left\{ \gamma'' \middle| \gamma'' = \langle \boldsymbol{a} \otimes \phi(\boldsymbol{o}), \boldsymbol{a} \otimes \phi(\boldsymbol{m}_i), \boldsymbol{a} \otimes \phi(\boldsymbol{n}_j) \rangle, i, j \in \{1, 2, \cdots, r\} \right\} \\ \bar{\mathcal{T}_a} &= \left\{ \bar{\gamma}'' \middle| \bar{\gamma}'' = \langle \bar{\boldsymbol{a}} \otimes \phi(\boldsymbol{o}), \bar{\boldsymbol{a}} \otimes \phi(\boldsymbol{m}_i), \bar{\boldsymbol{a}} \otimes \phi(\boldsymbol{n}_j) \rangle, i, j \in \{1, 2, \cdots, r\} \right\} \end{aligned} \tag{9}$$

where $\otimes$ denotes element-wise product and $\bar{\boldsymbol{a}} = 1 - \boldsymbol{a}$ denotes reverse attention.

Attention-guided triplets are obtained by highlighting important dimensions. Reverse attention handles the triplets in an opposite manner, which sheds light on these unimportant dimensions. We use reverse attention to check these neglected dimensions and render the attention module to find important dimensions as comprehensive as possible.

4.2.4 Triplet Deviation-Based Loss Function. The previous components are learnable after setting an objective. The queried outlier and its normal counterparts in triplets are expected to be separated clearly in the embedding space with attached attention. Data objects reversely masked by attention coefficient should be indistinguishable. Parameters in the ATON network are optimized to achieve these two targets. We define the loss of transferred triplet γ'' as follows:

$$L = \alpha \sum_{\gamma'' \in \mathcal{T}_a} \lambda(\gamma'') + (1 - \alpha) \sum_{\bar{\gamma}'' \in \bar{\mathcal{T}_a}} \bar{\lambda}(\bar{\gamma}'') \tag{10}$$

where $\lambda(\gamma'')$ is the loss term of the triplet masked by attention coefficient, while $\bar{\lambda}(\bar{\gamma}'')$ represents the loss term of the reversely masked triplet.

The concept of triplet loss [29] is employed to calculate the loss term $\lambda(\gamma'')$:

$$\begin{aligned}\lambda(\gamma'') &= \lambda\big(\langle \boldsymbol{o}^{\text{attn}}, \boldsymbol{m}_i^{\text{attn}}, \boldsymbol{n}_j^{\text{attn}}\rangle\big)\\ &= \max\big(d(\boldsymbol{m}_i^{\text{attn}}, \boldsymbol{n}_j^{\text{attn}}) - d(\boldsymbol{m}_i^{\text{attn}}, \boldsymbol{o}^{\text{attn}}) + e, 0\big)\end{aligned} \tag{11}$$

where $\langle \boldsymbol{o}^{\text{attn}}, \boldsymbol{m}_i^{\text{attn}}, \boldsymbol{n}_j^{\text{attn}}\rangle = \langle \boldsymbol{a}\otimes\phi(\boldsymbol{o}), \boldsymbol{a}\otimes\phi(\boldsymbol{m}_i), \boldsymbol{a}\otimes\phi(\boldsymbol{n}_j)\rangle$, $d(\cdot,\cdot)$ represents the Euclidean distance, and e is the margin.

This loss term is minimized to push $d(\boldsymbol{m}_i^{\text{attn}}, \boldsymbol{n}_i^{\text{attn}})$ to be smaller and $d(\boldsymbol{m}_i^{\text{attn}}, \boldsymbol{o}^{\text{attn}})$ to be larger than $d(\boldsymbol{m}_i^{\text{attn}}, \boldsymbol{n}_i^{\text{attn}})+e$, which means attention-guided embedding space is optimized to make the queried outlier be isolated from the normal data. It is noteworthy that this triplet loss term uses a relative concept of separability rather than an absolute manner. It provides a referenced distance between normal data, which can help ATON judges whether the queried outlier is effectively distinguished from normal data more accurately.

The loss term of reverse-attention-guided triplet $\bar{\lambda}(\bar{\gamma}_i'')$ is defined as follows:

$$\begin{aligned}\bar{\lambda}(\bar{\gamma}'') &= \bar{\lambda}\big(\langle \boldsymbol{o}^{\text{r-attn}}, \boldsymbol{m}_i^{\text{r-attn}}, \boldsymbol{n}_j^{\text{r-attn}}\rangle\big)\\ &= \big|d(\boldsymbol{m}_i^{\text{r-attn}}, \boldsymbol{n}_j^{\text{r-attn}}) - d(\boldsymbol{m}_i^{\text{r-attn}}, \boldsymbol{o}^{\text{r-attn}})\big|\end{aligned} \tag{12}$$

where $\langle \boldsymbol{o}^{\text{r-attn}}, \boldsymbol{m}_i^{\text{r-attn}}, \boldsymbol{n}_j^{\text{r-attn}}\rangle = \langle \bar{\boldsymbol{a}}\otimes\phi(\boldsymbol{o}), \bar{\boldsymbol{a}}\otimes\phi(\boldsymbol{m}_i), \bar{\boldsymbol{a}}\otimes\phi(\boldsymbol{n}_j)\rangle$.

This loss term is optimized to narrow the difference between distances of outlier-normal and normal-normal data objects within reverse-attention-guided triplets. We use this loss term to guarantee that the dimensions receiving less attention in the customized self-attention learning module are unimportant indeed. These dimensions should have very limited contributions to the separability of the queried outlier. In other words, the self-attention learning module can be forced to capture important dimensions as comprehensive as possible.

The coefficient of the loss function, i.e., α, is a hyper-parameter of ATON to adjust the weight on these two terms. We normally set α larger than 0.5 because the main objective is to learn a new space that can effectively separate the target outlier from its normal counterparts. In an extreme case, if α is 0, ATON will give all the dimensions with full attention to decrease the loss term $\bar{\lambda}(\bar{\gamma}'')$. We further detailedly examine the effect of α in Section 5.5.

ATON is optimized by Adam optimizer by adjusting parameters in layers of the feature embedding module and the customized self-attention learning module to minimize the loss function. Epoch number and batch size of the learning process are hyper-parameters of ATON, which are tested in Section 5.5. We can finally obtain an attention-guided embedding space that the queried outlier can be clearly separated from the normality.

4.2.5 Distilling Interpretation from ATON. To achieve the goal of outlier interpretation, we further distill the importance of original features from the embedding module and the customized self-attention learning module.

The weight vector of the original feature space $\boldsymbol{p} \in \mathbb{R}^D$ is computed as:

$$\boldsymbol{p} = |\mathbf{W}^{*\top}|\boldsymbol{a}^* \tag{13}$$

where $\mathbf{W}^{*\top}$ represents the transpose of the optimized weight matrix in the feature embedding module, $\boldsymbol{a}^*$ is the optimized attention coefficient vector, and $|\cdot|$ denotes that each element is transformed to absolute value in a given matrix. Note that the coefficients in $\mathbf{W}^*$ can be positive or negative values which indicate the impacts of original features to embedding dimensions from different directions. And the absolute values of elements in $\mathbf{W}^*$ represent the scale of impacts. Therefore, we use absolute value to measure their impact. If the absolute value of transformation coefficient of one feature is large (this coefficient is either positive or negative), a little change of the value of this feature will lead to a great change in embedding space. After getting attention value $\boldsymbol{a}^*$ (importance of embedding features) and the absolute value of embedding transformation matrix $|\mathbf{W}^*|$ (impact of original features to embedding space), Equation 13 is to transfer attention weights of the embedding space back to the original space.

We use ATON′ to represent ATON combined with a proposed threshold setting approach. ATON′ yields a feature subspace $\mathcal{E}$ as outlier interpretation result. We choose a feature subspace with the smallest size but the weight summation is larger than a threshold, which is represented as:

$$\mathcal{E} = \underset{\mathcal{E}\subset\mathcal{F},\ \sum_{f\in\mathcal{E}} \boldsymbol{p}(f)>t}{\arg\min} |\mathcal{E}| \tag{14}$$

where $t = \sqrt{\frac{2}{D}}\sum_{f\in\mathcal{F}}\boldsymbol{p}(f)$ is the threshold, and $|\mathcal{E}|$ is the size of the feature subspace. We use $\sqrt{\frac{2}{D}}$ as the threshold ratio, which is negatively correlated with the original space dimensionality. All the features will be retained if the full dimension is 2. The threshold ratio decreases with the increasing of the original dimensionality. Fixed threshold ratio will result in a huge feature subspace if the original data is in high-dimensional space, and thus we use the fraction $\frac{2}{D}$ to obtain a decreasing curve. We still maintain a relatively higher weight summation of interpretation subspace in high-dimensional data by employing a square root function.

4.3 Algorithm of ATON

Algorithm 1 presents the procedure of ATON. For simplicity, we do not explicitly present batch training process. In practical usage, ATON can be trained by setting proper epoch number and batch size. Steps (1-3) sample data into two candidate sets and generate a set of triplets $\mathcal{T}$. The network is trained in Steps (4-12). Step (6) converts the original features to a new space by linear transformation (parameterized by matrix $\mathbf{W}$). Customized self-attention learning module (parameterized by $\mathbf{W}_a$ and $\mathbf{W}'_a$) is in Step(7-9), which learns attention coefficient vector $\boldsymbol{a}$ and obtains attention-guided triplets $\mathcal{T}_a$ and reverse-attention-guided triplets $\bar{\mathcal{T}_a}$. Step (10-11) compute loss L and optimizes the network. The feature weight vector is distilled from the network in Step (13). A feature subspace is further obtained in Step (14). ATON finally returns $\boldsymbol{p}$ and $\mathcal{E}$ in Step (15).

We then analyze time complexity of some key steps. The calculation of finding the neighbor data in Step (1) incurs $O(N \times D \times r)$. Triplet set is with r^2 cardinality. Step (6) takes $O(n\_epochs \times r^2 \times D \times d)$. Attention module takes $O(n\_epochs \times r^2 \times (3dh_1 + h_1 d))$ in Step (8). The complexity of loss function in Step (10) is $O(n\_epochs \times r^2 \times d)$. Hyper-parameter r is normally set below 50, and d is 64

Algorithm 1 *ATON*($\mathcal{X}$, $\boldsymbol{o}$)

```
Input: $\mathcal{X}$ - data set, $\boldsymbol{o}$ - queried outlier
Output: $\boldsymbol{p}$ - interpretation feature weight vector, $\mathcal{E}$ - interpretation
    feature subspace
Select neighbor $r$ data objects of $\boldsymbol{o}$ into $\mathcal{X}_{\text{neighbor}}$
Randomly sample $r$ data objects into $\mathcal{X}_{\text{random}}$
$\mathcal{T} \leftarrow \{\gamma = \langle \boldsymbol{o}, \boldsymbol{m}, \boldsymbol{n} \rangle | \forall \boldsymbol{m} \in \mathcal{X}_{\text{random}}, \forall \boldsymbol{n} \in \mathcal{X}_{\text{neighbor}}\}$
Initialize $\mathbf{W} \in \mathbb{R}^{d \times D}$, $\mathbf{W}_a \in \mathbb{R}^{h_1 \times 3d}$, $\mathbf{W}'_a \in \mathbb{R}^{d \times h_1}$
repeat
    $\mathcal{T}_e \leftarrow \{\gamma' = \langle \mathbf{W}\boldsymbol{o}, \mathbf{W}\boldsymbol{m}_i, \mathbf{W}\boldsymbol{n}_j \rangle | i, j \in \{1, 2, \cdots, r\}\}$
    Flatten triplets to get matrix $\mathbf{U}$ by Equation (6)
    $\mathbf{A} \leftarrow \delta(\mathbf{W}'_a(\sigma(\mathbf{W}_a\mathbf{U})))$
    $\boldsymbol{a} \leftarrow \frac{1}{r^2} \sum_{i=1}^{d} \mathbf{A}(\cdot, i)$
    Generate $\mathcal{T}_a$ and $\bar{\mathcal{T}_a}$ by Equation (9)
    $L \leftarrow \alpha \sum_{\gamma''_i \in \mathcal{T}_a} \lambda(\gamma''_i) + (1 - \alpha) \sum_{\bar{\gamma}''_i \in \bar{\mathcal{T}_a}} \bar{\lambda}(\bar{\gamma}''_i)$
    Optimize network parameters by loss function $L$
until Reach maximum epoch number $n\_epochs$
$\boldsymbol{p} \leftarrow |\mathbf{W}^{*\top}| \boldsymbol{a}^*$
Convert $\boldsymbol{p}$ to $\mathcal{E}$ by Equation (14)
return $\boldsymbol{p}, \mathcal{E}$
```

or $\lfloor 1.5D \rfloor$. Thus, ATON has quadratic time complexity w.r.t. data dimensionality and linear time complexity w.r.t. data size.

5 EXPERIMENTS

In this section, we conduct experiments to answer the following questions:

- Effectiveness: How accurate are the outlier interpretation results computed by ATON and its contenders on real-world datasets?
- Case Studies: Can ATON produce meaningful interpretation results in typical cases?
- Ablation Study: Do key designs of ATON contribute to better interpretation results?
- Parameter Test: How do the hyper-parameters influence the interpretation performance of ATON?
- Scalability Test: Does ATON have good scalability compared to its competitors w.r.t. dimensionality and data size?

We first introduce the experimental setup before detailing our empirical findings.

5.1 Experimental Setup

5.1.1 Datasets. Twelve real-world outlier detection datasets are used in the experiments. Datasets *WineR* and *WineW* are short for *Wine Quality (red)* and *Wine Quality (white)*. These two datasets are from Kaggle platform (https://www.kaggle.com/). Other datasets are publicly-available at ODDS [26] (an outlier detection dataset library). All of these datasets are with real outliers or semantic outliers. We skip the detection process and perform model-agnostic outlier interpretation methods to explain each real outlier.

5.1.2 Competitors. ATON outputs a feature weight vector per queried outlier as explanation, and ATON′ further yields interpretation feature subspace by the proposed threshold setting approach. ATON and ATON′ are compared with two types of outlier interpretation methods (**Type-I:** feature weight as output; **Type-II:** feature subspace as output) respectively. We choose five competitors, including both outlier interpretation methods and general classifier explanation methods, which are introduced as follows:

- COIN and COIN′ [17]: COIN is a state-of-the-art outlier interpretation method that outputs a sparse feature weight vector indicating the abnormality of each feature. We further obtain feature subspace by only retaining features with positive weight and filtering features with zero weight, which is denoted as COIN′.
- SiNNE [28]: SiNNE is the newest state-of-the-art outlier interpretation method utilizing score-and-search manner. It directly outputs feature subspace for each queried outlier.
- SHAP [18] and LIME [27]: SHAP and LIME are classifier explanation methods that are commonly used in the field of interpretable machine learning. They explain a prediction by assigning each feature an importance value. They are used as Type-I competitors.

5.1.3 Parameter Settings and Implementations. ATON is performed by using sampling number $r = 30$, coefficient of loss function $\alpha = 0.8$, and embedding space dimension $d = 64$. The network is trained by 10 epochs and 64 triplets per batch. The hidden layer in the attention net has $1.5d$ hidden units. We use Adam optimizer with the 0.1 learning rate and apply a early stopping mechanism during the network training process. COIN uses default recommended parameters. In terms of SiNNE, the width of the beam search is set as 10, the ensemble number is 100, and the sampling number is 8. As for SHAP and LIME, we use SVM with RBF kernel as classifier.

All the outlier interpretation methods are implemented in Python. The source code of COIN is released by its original authors. We reimplement SiNNE. The classifier explanation methods SHAP and LIME are publicly-available Python packages.

5.1.4 Ground-Truth Annotations. Evaluating outlier interpretation task requires benchmark datasets with ground-truth annotations (ideal interpretation feature subspace for each outlier). To the best of our knowledge, there is no public-available real-world dataset with such annotations. To address this gap, we propose a new labeling method and release the ground-truth annotations. For a real-world dataset, we employ three different kinds of representative outlier detection methods (i.e., ensemble-based method iForest [16], probability-based method COPOD [15], and distance-based method HBOS [8]) to evaluate outlying degree of real outliers given every possible subspace. As defined in Section 3, a good explanation for an outlier should be a high-contrast subspace that the outlier explicitly demonstrates its outlierness, and outlier detectors can easily and certainly predict it as an outlier in this subspace. Therefore, the ground-truth interpretation for each outlier is defined as the subspace that the outlier obtains the highest outlier score among all the possible subspaces. Thus, we finally get three lists of ground truth annotations according to three selected outlier detectors. Note that the original dimensionality of some datasets is high. Such a brute-force searching process has exponential time complexity. The number of possible subsets increases to 32,768 when a dataset has 15 dimensions, which means outlier detectors

Table 1: Outlier Interpretation Performance of Type-I Methods. Three rows of each dataset represent interpretation performance using the ground-truth annotations generated by different outlier detectors (iForest, COPOD, and HBOS), respectively.

DATA	**Precision**				**Jaccard Index**				**AUPR**				**AUROC**			
	ATON	COIN	SHAP	LIME	ATON	COIN	SHAP	LIME	ATON	COIN	SHAP	LIME	ATON	COIN	SHAP	LIME
Pima	**0.705**	0.510	0.442	0.432	**0.604**	0.399	0.327	0.319	**0.793**	0.601	0.532	0.525	**0.836**	0.702	0.597	0.597
	0.666	0.466	0.508	0.502	**0.560**	0.362	0.397	0.387	**0.763**	0.543	0.586	0.587	**0.788**	0.633	0.653	0.657
	0.600	0.246	0.489	0.421	**0.537**	0.198	0.430	0.351	**0.680**	0.348	0.571	0.520	**0.782**	0.572	0.715	0.693
Vertebral	**0.692**	0.499	0.389	0.392	**0.655**	0.411	0.318	0.324	**0.767**	0.556	0.435	0.469	**0.814**	0.597	0.654	0.649
	0.741	0.379	0.528	0.583	**0.676**	0.301	0.458	0.512	**0.802**	0.517	0.538	0.601	**0.873**	0.537	0.644	0.666
	0.507	0.270	0.083	0.177	**0.493**	0.253	0.078	0.173	**0.597**	0.341	0.240	0.318	**0.732**	0.451	0.507	0.527
WineR	**0.665**	0.528	0.384	0.304	**0.561**	0.422	0.273	0.213	**0.757**	0.608	0.361	0.374	**0.872**	0.726	0.614	0.564
	0.648	0.466	0.436	0.388	**0.535**	0.337	0.312	0.271	**0.741**	0.558	0.398	0.428	**0.851**	0.680	0.661	0.632
	0.496	0.446	0.348	0.389	**0.428**	0.378	0.277	0.326	**0.599**	0.503	0.379	0.428	**0.764**	0.714	0.707	0.651
WineW	**0.667**	0.485	0.473	0.348	**0.576**	0.395	0.377	0.262	**0.743**	0.551	0.534	0.421	**0.853**	0.731	0.658	0.622
	0.631	0.546	0.456	0.373	**0.505**	0.422	0.324	0.259	**0.718**	0.616	0.511	0.404	**0.799**	0.696	0.609	0.552
	0.525	0.291	0.473	0.388	**0.467**	0.233	0.412	0.325	**0.612**	0.385	0.545	0.466	**0.800**	0.646	0.642	0.612
Heart	**0.846**	0.753	0.669	0.634	**0.778**	0.676	0.580	0.554	**0.899**	0.797	0.739	0.704	**0.901**	0.836	0.695	0.676
	0.809	0.699	0.432	0.412	**0.736**	0.628	0.294	0.290	**0.841**	0.757	0.528	0.509	**0.878**	0.798	0.542	0.535
	0.784	0.724	0.554	0.546	**0.725**	0.653	0.489	0.479	**0.834**	0.780	0.602	0.577	**0.872**	0.825	0.611	0.608
Ionosphere	**0.637**	0.458	0.425	0.299	**0.533**	0.354	0.312	0.213	**0.746**	0.589	0.497	0.400	**0.836**	0.734	0.674	0.558
	0.682	0.568	0.554	0.495	**0.558**	0.427	0.424	0.368	**0.767**	0.635	0.635	0.578	**0.835**	0.681	0.735	0.718
	0.648	0.546	0.482	0.351	**0.561**	0.456	0.389	0.267	**0.706**	0.657	0.579	0.422	**0.846**	0.778	0.747	0.631
letter	**0.687**	0.539	0.619	0.492	**0.563**	0.405	0.495	0.379	**0.769**	0.589	0.653	0.550	**0.794**	0.687	0.704	0.591
	0.688	0.490	0.579	0.507	**0.560**	0.347	0.435	0.372	**0.768**	0.500	0.549	0.483	**0.816**	0.642	0.671	0.592
	0.636	0.433	0.606	0.415	**0.570**	0.363	0.536	0.349	**0.703**	0.529	0.657	0.484	**0.802**	0.762	0.773	0.619
Arrhythmia	**0.741**	0.179	0.651	0.401	**0.673**	0.122	0.566	0.303	**0.817**	0.256	0.738	0.460	**0.916**	0.451	0.853	0.728
	0.567	0.207	0.514	0.440	**0.478**	0.144	0.417	0.337	**0.664**	0.297	0.599	0.492	**0.828**	0.464	0.731	0.722
	0.692	0.119	0.659	0.364	**0.655**	0.093	0.634	0.323	**0.756**	0.206	0.705	0.447	**0.919**	0.459	0.865	0.768
WBC	**0.695**	0.529	0.502	0.352	**0.597**	0.430	0.408	0.263	**0.734**	0.602	0.566	0.419	**0.869**	0.717	0.715	0.655
	0.569	0.356	0.425	0.389	**0.480**	0.257	0.321	0.294	**0.671**	0.508	0.530	0.498	**0.830**	0.634	0.691	0.684
	0.614	0.564	0.548	0.558	**0.526**	0.495	0.468	0.470	**0.700**	0.635	0.631	0.586	**0.882**	0.780	0.822	0.770
Satimage	**0.637**	0.541	0.636	0.488	**0.515**	0.424	0.512	0.374	0.674	0.610	**0.686**	0.567	0.772	0.705	**0.799**	0.672
	0.691	0.531	0.545	0.483	**0.554**	0.398	0.405	0.351	**0.727**	0.610	0.643	0.576	**0.834**	0.674	0.717	0.631
	0.792	0.730	0.792	0.685	**0.764**	0.706	0.761	0.658	**0.836**	0.776	0.807	0.723	**0.943**	0.901	0.887	0.834
Speech	**0.724**	0.513	0.668	0.552	**0.615**	0.407	0.552	0.417	**0.813**	0.580	0.771	0.650	**0.851**	0.714	0.776	0.660
	0.672	0.579	0.653	0.581	**0.545**	0.448	0.516	0.434	0.776	0.624	**0.777**	0.695	**0.814**	0.730	0.783	0.666
	0.731	0.368	0.645	0.470	**0.691**	0.318	0.600	0.410	**0.794**	0.468	0.712	0.557	**0.927**	0.765	0.817	0.697
Optdigits	**0.699**	0.631	0.686	0.677	**0.568**	0.488	0.555	0.544	**0.816**	0.690	0.814	0.787	0.807	0.741	**0.809**	0.768
	0.708	0.605	0.634	0.669	**0.575**	0.461	0.488	0.538	**0.825**	0.647	0.767	0.766	**0.795**	0.719	0.727	0.729
	0.817	0.452	**0.869**	0.795	0.772	0.414	**0.834**	0.748	0.863	0.580	**0.910**	0.850	0.913	0.886	**0.942**	0.903
Average	**0.675**	0.479	0.538	0.465	**0.589**	0.390	0.444	0.374	**0.752**	0.557	0.603	0.537	**0.840**	0.688	0.715	0.662
Improvement	-	40.9%	25.5%	45.2%	-	51.0%	32.7%	57.5%	-	35.0%	24.7%	40.0%	-	22.1%	17.5%	26.9%

need to perform 32,768 times on a single dataset. Therefore, we employ PCA on datasets with over 15 features to extract 10 features. The labeling procedure and subsequent experiments are conducted on these processed datasets.

5.1.5 *Performance Evaluation Metrics.*

After getting the ground truth explanation, we can quantitatively measure the performance of different outlier interpretation methods.

Type-II interpretation methods directly output feature subspace for each queried outlier, they can be compared without any further processing. We utilize three evaluation metrics, i.e., precision, recall, and F_1 score to evaluate the quality of interpretation subspaces. Let the ground-truth subspace as $\mathcal{G}$ and the predicted subspace as $\mathcal{P}$, precision is defined as $|\mathcal{G} \cap \mathcal{P}|/|\mathcal{P}|$, recall is $|\mathcal{G} \cap \mathcal{P}|/|\mathcal{G}|$, and F_1 score is harmonic mean of precision and recall, i.e., $F_1 = 2(\text{precision} \times \text{recall})/(\text{precision} + \text{recall})$.

In terms of the comparison between Type-I methods that output feature weight, we need to transfer the computed weight vector to interpretation feature subspace by gathering top-ranked features with large weight into a subset. For simplicity, the subset size is the same as the real length of the ground-truth subspace. It is a fair competition because all the Type-I interpretation methods are incorporated with this transformation process. We employ precision and Jaccard index to evaluate the transferred subspace. Note that recall is equal to precision if we assume the ground-truth subspace and the predicted subspace are with the same size, and thus we only use precision here. Jaccard index is frequently utilized to evaluate the similarity of two sets, which is calculated as $|\mathcal{G} \cap \mathcal{P}|/|\mathcal{G} \cup \mathcal{P}|$. Type-I interpretation methods can yield a feature ranking according to the computed weight vector. Hence, to directly and impartially evaluate the ranking quality without determining the size of explanation subspace, we further employ the Area under the Precision-Recall curve (AUPR) and the Area under the ROC curve (AUROC).

All of these metrics are computed to evaluate the interpretation subspace of each queried outlier, and we report the average interpretation performance of all the outliers in each dataset. These metrics range from 0 to 1, and higher values indicate better performance.

Table 2: Outlier Interpretation Performance of Type-II Methods. Three rows of each dataset denote interpretation performance using different ground-truth annotations. Imp. is short for average improvement rate of F_1 score.

DATA	**F_1 score (Precision / Recall)**		
	ATON′	SiNNE	COIN′
Pima	**0.673** (0.627 / 0.727)	0.588 (0.546 / 0.637)	0.553 (0.384 / 0.985)
	0.650 (0.618 / 0.687)	0.557 (0.530 / 0.588)	0.586 (0.417 / 0.986)
	0.531 (0.420 / 0.721)	0.441 (0.347 / 0.604)	0.415 (0.262 / 0.995)
Vertebral	**0.628** (0.520 / 0.793)	0.597 (0.512 / 0.716)	0.468 (0.307 / 0.987)
	0.703 (0.593 / 0.864)	0.565 (0.492 / 0.662)	0.524 (0.355 / 0.998)
	0.406 (0.283 / 0.720)	0.389 (0.280 / 0.637)	0.329 (0.197 / 0.997)
WineR	**0.661** (0.582 / 0.765)	0.505 (0.443 / 0.588)	0.429 (0.297 / 0.777)
	0.652 (0.606 / 0.706)	0.493 (0.459 / 0.533)	0.450 (0.323 / 0.741)
	0.481 (0.379 / 0.655)	0.361 (0.294 / 0.466)	0.408 (0.265 / 0.895)
WineW	**0.619** (0.534 / 0.737)	0.531 (0.444 / 0.660)	0.436 (0.287 / 0.903)
	0.605 (0.609 / 0.601)	0.528 (0.518 / 0.538)	0.497 (0.357 / 0.819)
	0.479 (0.369 / 0.684)	0.388 (0.292 / 0.579)	0.380 (0.236 / 0.977)
Heart	**0.730** (0.659 / 0.817)	0.738 (0.694 / 0.787)	0.619 (0.510 / 0.787)
	0.770 (0.763 / 0.778)	0.578 (0.597 / 0.561)	0.664 (0.592 / 0.757)
	0.656 (0.563 / 0.786)	0.630 (0.556 / 0.726)	0.576 (0.452 / 0.792)
Ionosphere	0.622 (0.520 / 0.775)	0.482 (0.457 / 0.512)	**0.629** (0.613 / 0.646)
	0.671 (0.644 / 0.700)	0.454 (0.523 / 0.401)	0.573 (0.643 / 0.517)
	0.618 (0.507 / 0.793)	0.433 (0.414 / 0.455)	**0.647** (0.610 / 0.690)
Letter	0.665 (0.668 / 0.663)	**0.668** (0.698 / 0.641)	0.562 (0.398 / 0.960)
	0.664 (0.672 / 0.656)	0.614 (0.649 / 0.583)	0.554 (0.390 / 0.951)
	0.545 (0.446 / 0.698)	**0.616** (0.513 / 0.771)	0.403 (0.254 / 0.977)
Arrhythmia	**0.676** (0.551 / 0.875)	0.564 (0.448 / 0.763)	0.367 (0.225 / 1.000)
	0.596 (0.507 / 0.724)	0.499 (0.417 / 0.621)	0.398 (0.249 / 0.999)
	0.557 (0.401 / 0.910)	0.473 (0.333 / 0.814)	0.273 (0.158 / 1.000)
WBC	**0.604** (0.482 / 0.807)	0.570 (0.464 / 0.740)	0.560 (0.532 / 0.591)
	0.601 (0.516 / 0.719)	0.449 (0.390 / 0.531)	0.461 (0.494 / 0.432)
	0.579 (0.445 / 0.829)	0.502 (0.393 / 0.695)	**0.639** (0.580 / 0.710)
Satimage	**0.585** (0.520 / 0.669)	0.429 (0.463 / 0.400)	0.429 (0.273 / 1.000)
	0.644 (0.659 / 0.629)	0.410 (0.534 / 0.332)	0.539 (0.369 / 1.000)
	0.541 (0.379 / 0.945)	0.442 (0.348 / 0.606)	0.247 (0.141 / 1.000)
Speech	0.693 (0.690 / 0.697)	**0.718** (0.707 / 0.730)	0.525 (0.356 / 1.000)
	0.653 (0.695 / 0.615)	**0.654** (0.683 / 0.628)	0.549 (0.378 / 1.000)
	0.615 (0.476 / 0.871)	0.626 (0.486 / 0.879)	0.342 (0.206 / 1.000)
Optdigits	**0.671** (0.759 / 0.601)	0.654 (0.732 / 0.591)	0.607 (0.437 / 0.995)
	0.672 (0.749 / 0.610)	0.662 (0.727 / 0.607)	0.593 (0.423 / 0.992)
	0.557 (0.409 / 0.873)	**0.580** (0.430 / 0.892)	0.298 (0.175 / 1.000)
Average	**0.619** (0.551 / 0.742)	0.539 (0.495 / 0.624)	0.487 (0.365 / 0.885)
Imp.	-	14.8%	27.1%

5.2 Effectiveness of Interpretation

5.2.1 Settings. Seven outlier interpretation methods (i.e., ATON, ATON′, COIN, COIN′, SiNNE, SHAP, and LIME) are performed on twelve real-world datasets. As introduced in Section 5.1.5, Type-I methods are evaluated using four metrics (precision, Jaccard index, AUPR, and AUROC), while Type-II methods are estimated by precision, recall, and F_1 score. We independently execute these methods ten times on each dataset and report the average performance.

5.2.2 Results and Analysis. Table 1 and Table 2 show the interpretation performance of Type-I and Type-II methods, respectively. The best performance on each dataset is highlighted in bold. ATON is the best performer on nine out of twelve datasets according to four evaluation metrics and the ground-truth annotations generated by three outlier detectors. F_1 score of ATON′ is higher than all of its contenders on seven datasets. Averagely, ATON and ATON′ produce significant performance leap over the state-of-the-art outlier interpretation methods and classifier explanation methods.

ATON can generate more accurate interpretation results on real-world datasets, i.e., outliers can be better identified by different kinds of outlier detectors when using corresponding interpretation subspaces. State-of-the-art outlier interpretation method COIN and COIN′ [17] fail to obtain sufficient results. COIN′ tends to produce very large interpretation subspaces and obtain a high recall, but the generated subspace is mixed with many irrelevant features, i.e., its precision is very low. The overall performance F_1 of COIN′ is still inferior compared to the proposed ATON. Note that the original work of COIN [17] conducts experiments by first appending multiple noise features and simply assuming all the original features as correct interpretation results, which is different from our settings. This experiment setting is somewhat reasonable when the ground-truth interpretation subspace is unavailable. Arguably, these noise features can be easily identified because they do not have any correlation with the original features. Thus, COIN is ineffective in these practical situations. We also analyze its possible flaws in Section 2. SiNNE employs score-and-search manner, which means it can only produce suboptimal results. It is surprising to find that the classifier explanation method SHAP can obtain comparably good performance. It might be due to the similar nature between classifier explanation and outlier interpretation (i.e., both of them investigate the effect of features to separate the queried outlier with other normal data). SHAP is a very powerful and well-known method to explain the classifier predictions. Nevertheless, our method still has advantages to handle outlier interpretation task.

It is noteworthy that all the outlier interpretation methods cannot yield very accurate performance (e.g., over 0.9 precision or Jaccard index). It is mainly due to two reasons: (i) Outlier interpretation is a non-trivial task. It is very challenging to capture completely correct feature subspace, especially in complex real-world datasets. The ground-truth subspace normally has no more than five features, i.e., the precision decreases to only 0.6 when interpretation approaches incorrectly retrieve two features; and (ii) We use three outlier detectors to imitate human analysts. Different outlier detectors may favor inconsistent feature subspace as interpretation. The ground-truth annotations generated by outlier detectors might still be unable to fully represent the real interpretations in practical scenarios. These datasets are from various domains, i.e., proper domain knowledge should be considered when interpreting outliers.

5.3 Case Studies

5.3.1 Settings: This experiment aims to visualize the effect of different outlier interpretation methods and further illustrates the effectiveness of ATON through some case studies. We employ MNIST dataset in this experiment. MNIST is a popular image dataset containing handwritten digits in 28×28 pixels. We first flatten the

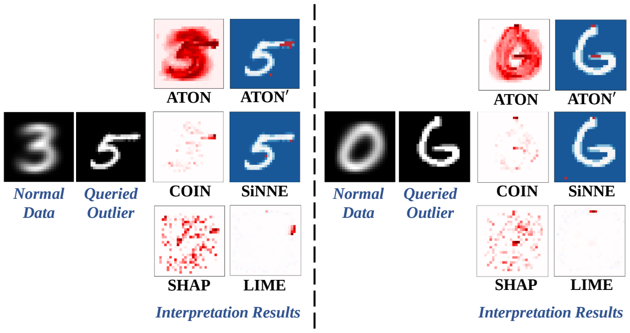

Figure 3: Interpretation Results of Two Outlier Images (Number 5 and 6) from MNIST Dataset. Results of Type-I outlier interpretation methods are shown in white background. Darker red pixels indicate higher interpretation feature weight. Type-II interpretation results are in blue background. The interpretation subspace is highlighted in red.

images and obtain vectors with 784 dimensions. It is then preprocessed by setting one class as normal data and sampling 20 images from the other class as anomalies. Two imbalanced datasets are then generated by choosing relatively similar numbers as normal and anomaly data, i.e., 3 vs. 5 and 0 vs. 6. In this experiment, ATON is performed by setting the hyper-parameter *d* (dimension of the embedding space) as 512 because these cases are with relatively high original dimensions. Other hyper-parameters are kept the same as reported configurations in the experimental setup section.

5.3.2 Results and Analysis. Figure 3 shows the interpretation results on MNIST. We also present the normal pattern image (average value of all the normal data) and the queried outlier image using a black background. The results of COIN′ are omitted because they can be directly obtained by using a single red color to tint on the dark red pixels in the image of COIN. ATON successfully highlights all the pixels that the outlier behaves differently compared to the normal pattern. The most different part, e.g., the right-most part of number 5, is given the highest weight, as shown in the interpretation image of ATON′. Note that the red color in digit 3's left-hand positions in ATON's result indicates that the outlier digit does not pass these pixels but these positions should have trace in the normal condition, which means these pixels are also important to distinguish digit 5 and digit 3. LIME and COIN can also find the correct part, but SHAP cannot produce meaningful results. SiNNE is also ineffective and inefficient in high-dimensional data. It only reports two pixels.

5.4 Ablation Study

5.4.1 Settings. This section is to validate the significance of three key components of ATON. ATON uses the customized self-attention learning module to learn contributions of embedding dimensions. We remove this component and only use triplet deviation to optimize the feature embedding module. This variant is to corroborate whether the attention module brings better interpretation results, which is denoted as Abla-I. To test the effect of the feature embedding module, the first linear transformation layer is removed, and ATON directly learns attention of the original space. This ablated version is denoted as Abla-II. Besides, the triplet deviation-based loss function is replaced with a multi-layer perceptron classifier and cross-entropy loss in ablated version Abla-III. We use Abla-III to investigate the contribution of triplet deviation loss. Other components of these variants stay the same with ATON.

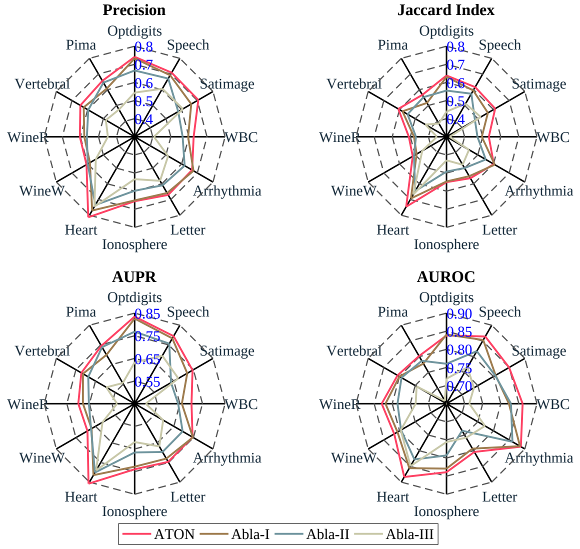

Figure 4: Ablation Study Results. Outlier interpretation performance (precision, Jaccard Index, AUPR, and AUROC) of ATON and its three ablated versions on twelve datasets.

Table 3: Average Performance of Ablation Study and Improvement Rates of ATON over its Ablated Versions

Method	Precision	Jaccard Index	AUPR	AUROC
ATON	0.675	0.589	0.752	0.840
Abla-I	0.653 (3.4%)	0.563 (4.4%)	0.731 (2.9%)	0.824 (1.9%)
Abla-II	0.621 (8.7%)	0.529 (11.2%)	0.701 (7.3%)	0.799 (5.1%)
Abla-III	0.541 (24.8%)	0.453 (29.8%)	0.622 (20.9%)	0.751 (11.91%)

5.4.2 Results and Analysis. The interpretation performance of ATON and its three ablated versions is shown in Figure 4. Each method corresponds to one circle in Figure 4, indicating the average value over the results evaluated by three annotation lists on twelve datasets. A summary of the average performance is presented in Table 3. We report the average performance of all the datasets and the improvement rate of ATON over its three ablated variants. ATON prevails Abla-I on almost all the twelve datasets, which obtains noteworthy improvement. ATON outperforms Abla-II and Abla-III on all the datasets and achieves more significant performance improvement.

This experiment validates that each component in ATON does contribute to better interpretation performance. It is surprising to find that Abla-I can achieve very good results by only using feature embedding module and triplet deviation-based loss function. This also indicates that the feature embedding module is important for outlier interpretation. Higher-level feature patterns and richer semantics can be explicitly unfolded in this embedding space, and thus the outlying behaviors of the queried outlier can be directly seized. The results of Abla-II also quantitatively validate the significance of the feature embedding module. Note that Abla-I has comparably or slightly better performance on two exceptions (datasets Arrhythmia and Optdigits) compared to ATON. It is because ATON might suffer from overfitting problem on these two datasets. Parameters of the attention module, e.g., hidden layer number and hidden unit number per layer, can be further adjusted to obtain more satisfactory performance in practical scenarios. Abla-I might have slight advantages on a few datasets thanks to its plain structure. However, ATON is still superior on most of the datasets, and the attention module is also proved to be an effective component. Triplet deviation-based loss function plays a fundamental role in the learning process of ATON. A powerful classifier can still well separate the queried outlier and the normality in a poor subspace, and thus Abla-III only obtains very inferior performance.

5.5 Parameter Test

5.5.1 Settings. We investigate the influence of different hyperparameter settings of ATON to its interpretation performance. There are five hyper-parameters in ATON, i.e., sampling number r, coefficient α in the loss function, dimensions of embedding space d, network training batch size, and training epoch number.

5.5.2 Results and Analysis. Parameter test results are shown in Figure 5. We report results on six representative datasets and the average performance over all the twelve datasets.

Generally, interpretation performance improves when sampling number r and embedding dimension d increase. Higher sampling number r directly leads to more abundant training data for the network. Rich normal samples can better represent the normality. Higher dimension means the new embedding space is with richer semantics. Each dimension can be seen as a pattern indicating one combination of the original features. ATON can accordingly generate more reliable interpretation with the help of a large number of patterns. Nevertheless, the improvement of interpretation quality is limited when r reaches 30 and d reaches 64. Improving normal samples and embedding dimension can only bring bounded gain. The normality might be fully represented with 30 sampling number. 64-dimension linear layer is powerful enough as these processed datasets are with only less than 15 features.

As for coefficient α in the loss function, different datasets have inconsistent situations when changing α, whereas ATON is normally robust to this parameter (the fluctuation is only within 0.05). It is recommended to set α as 0.8 for most of the datasets. The loss function can consider both attention-guided triple-wise distance and the behavior of triplets with opposite attention. It is better to attach more importance to the attention-guided element but totally ignoring the opposite attention is still inferior on some datasets.

In terms of the batch size and the epoch number, ATON performs stably w.r.t. different configurations. The training data is 900 triplets when the sampling number is 30. It is better to choose a larger batch size, say 512, to guide the network to reach the global optima. Empirically, ATON can be well trained by 10 epochs. The performance is stable because we adopt the early stopping mechanism in ATON.

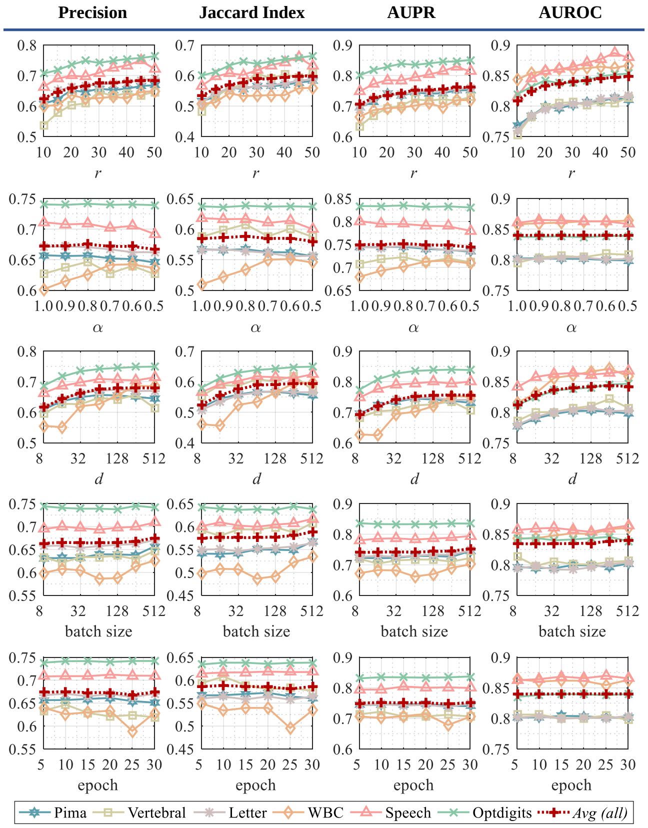

Figure 5: Parameter Test Results. Outlier interpretation performance (precision, Jaccard Index, AUPR, and AUROC) of ATON with different hyper-parameters (r, α, d, batch size, and epoch number).

5.6 Scalability Test

5.6.1 Settings. We create a group of synthetic datasets to evaluate the scalability of different outlier interpretation methods w.r.t. data dimensionality and data size. Five datasets are generated with varying data dimensions ({8, 32, 128, 512, 2,048}) and fixed data size (1,000). Another five datasets are with different data size ({1,000, 4,000, 16,000, 64,000, 256,000}) and the same data dimensionality (32). In these datasets, outliers account for 0.5% of the whole dataset.

5.6.2 Results and Analysis. Figure 6 shows the scalability test results w.r.t. data dimensionality and data size. ATON presents outstanding scalability compared to other outlier interpretation methods. Our method runs approximate two magnitudes faster than SHAP and LIME on high-dimensional datasets. Neural network can efficiently handle high-dimensional data with the help of the development of GPU. The competitor COIN has comparably fast execution speed w.r.t. data dimensionality, but SiNNE runs out of memory on the dataset with 2,048 dimensions. In terms of the scale-up test w.r.t. data size, ATON, SHAP, and LIME have linear time complexity. ATON needs the shortest execution time because it utilizes sampling in the triplet generator. COIN fail to return a result within two days on the dataset containing 64,000 data objects.

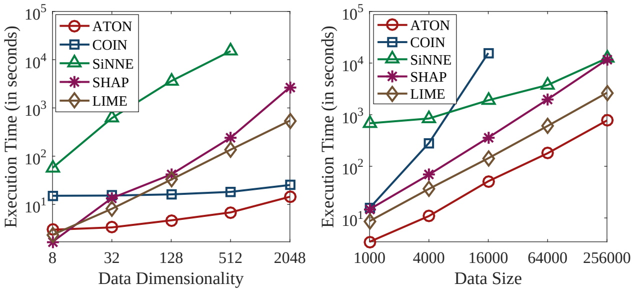

Figure 6: Scalability Test Results w.r.t. Data Dimensionality and Data Size. SiNNE runs out of memory on the dataset with 2,048 dimensions. COIN cannot output a result within two days on the dataset with 64,000 data objects.

6 CONCLUSION AND FUTURE WORK

This paper addresses the problem of how to explain outliers detected by any black-box outlier detector. We introduce ATON, a novel attention-guided triplet deviation network, which is model-agnostic and domain-agnostic. Instead of following the popular subspace searching manner, ATON directly learns an optimal embedding space with attached attention to better seize the outlierness of the queried outlier, leading to more accurate interpretation results. Extensive experiments show that ATON achieves significant performance improvement over the state-of-the-art outlier interpretation methods and the general classifier explanation methods on real-world datasets. ATON can give meaningful interpretation results in visualized cases. ATON also obtains outstanding scalability compared to its competitors. In the future, we plan to apply ATON to failure diagnosis of JointCloud service systems [32].

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2016YFB1000101), the National Natural Science Foundation of China (No. 62002371, No.61379052, and No.61472439), the Science Foundation of Ministry of Education of China (No.2018A02002), and the Natural Science Foundation for Distinguished Young Scholars of Hunan Province (No.14JJ1026).

REFERENCES

[1] Charu C Aggarwal. 2017. *Outlier analysis*. Springer. https://doi.org/10.1007/978-1-4614-6396-2

[2] Fabrizio Angiulli, Fabio Fassetti, Giuseppe Manco, and Luigi Palopoli. 2017. Outlying property detection with numerical attributes. *Data mining and knowledge discovery* 31, 1 (2017), 134–163.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

[4] Xuan Hong Dang, Ira Assent, Raymond T Ng, Arthur Zimek, and Erich Schubert. 2014. Discriminative features for identifying and interpreting outliers. In *ICDE*. IEEE, 88–99.

[5] Xuan Hong Dang, Barbora Micenková, Ira Assent, and Raymond T Ng. 2013. Local outlier detection with interpretation. In *ECML PKDD*. Springer, 304–320.

[6] Lei Duan, Guanting Tang, Jian Pei, James Bailey, Akiko Campbell, and Changjie Tang. 2015. Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery* 29, 5 (2015), 1116–1151.

[7] Ioana Giurgiu and Anika Schumann. 2019. Additive Explanations for Anomalies Detected from Multivariate Temporal Data. In *CIKM*. 2245–2248.

[8] Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track* (2012), 59–63.

[9] Nikhil Gupta, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos. 2018. Beyond outlier detection: Lookout for pictorial explanation. In *ECML PKDD*. Springer, 122–138.

[10] Songlei Jian, Guansong Pang, Longbing Cao, Kai Lu, and Hang Gao. 2018. Cure: Flexible categorical data representation by hierarchical coupling learning. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (2018), 853–866.

[11] Jacob Kauffmann, Klaus-Robert Müller, and Grégoire Montavon. 2020. Towards explaining anomalies: a deep Taylor decomposition of one-class models. *Pattern Recognition* 101 (2020), 107198.

[12] Fabian Keller, Emmanuel Müller, Andreas Wixler, and Klemens Böhm. 2013. Flexible and adaptive subspace search for outlier analysis. In *CIKM*. 1381–1390.

[13] Martin Kopp, Tomáš Pevnỳ, and Martin Holeňa. 2020. Anomaly explanation with random forests. *Expert Systems with Applications* 149 (2020), 113187.

[14] Chia-Tung Kuo and Ian Davidson. 2016. A framework for outlier description using constraint programming. In *AAAI*. 1237–1243.

[15] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. 2020. COPOD: Copula-Based Outlier Detection. In *ICDM*. IEEE.

[16] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2012. Isolation-Based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data* 6, 1, Article 3 (2012), 39 pages.

[17] Ninghao Liu, Donghwa Shin, and Xia Hu. 2018. Contextual outlier interpretation. In *IJCAI*. 2461–2467.

[18] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *NeuralPS*. 4765–4774.

[19] Meghanath Macha and Leman Akoglu. 2018. Explaining anomalies in groups with characterizing subspace rules. *Data Mining and Knowledge Discovery* 32, 5 (2018), 1444–1480.

[20] Barbora Micenková, Raymond T Ng, Xuan-Hong Dang, and Ira Assent. 2013. Explaining outliers by subspace separability. In *ICDM*. IEEE, 518–527.

[21] Christoph Molnar. 2020. *Interpretable Machine Learning*. Lulu. com.

[22] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. 2018. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *SIGKDD*. 2041–2050.

[23] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2020. Deep learning for anomaly detection: A review. *arXiv preprint arXiv:2007.02500* (2020).

[24] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep anomaly detection with deviation networks. In *SIGKDD*. 353–362.

[25] Tomáš Pevnỳ and Martin Kopp. 2014. Explaining anomalies with sapling random forests. In *Information Technologies-Applications and Theory Workshops, Posters, and Tutorials*.

[26] Shebuti Rayana. 2016. ODDS Library. http://odds.cs.stonybrook.edu

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *SIGKDD*. ACM, 1135–1144.

[28] Durgesh Samariya, Kai Ming Ting, and Sunil Aryal. 2020. A new effective and efficient measure for outlying aspect mining. *arXiv preprint arXiv:2004.13550* (2020).

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*. 815–823.

[30] Md Amran Siddiqui, Alan Fern, Thomas G Dietterich, and Weng-Keen Wong. 2019. Sequential feature explanations for anomaly detection. *ACM Transactions on Knowledge Discovery from Data* 13, 1 (2019), 1–22.

[31] Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Jian Pei. 2016. Discovering outlying aspects in large datasets. *Data Mining and Knowledge Discovery* 30, 6 (2016), 1520–1555.

[32] Huaimin Wang, Peichang Shi, and Yiming Zhang. 2017. JointCloud: A Cross-Cloud Cooperation Architecture for Integrated Internet Service Customization. In *ICDCS*. IEEE, 1846–1855.

[33] Hongzuo Xu, Yongjun Wang, Li Cheng, Yijie Wang, and Xingkong Ma. 2018. Exploring a High-quality Outlying Feature Value Set for Noise-Resilient Outlier Detection in Categorical Data. In *CIKM*. ACM, 17–26.

[34] Hongzuo Xu, Yongjun Wang, Zhiyue Wu, and Yijie Wang. 2019. Embedding-based Complex Feature Value Coupling Learning for Detecting Outliers in Non-IID Categorical Data. In *AAAI*. AAAI Press, 5541–5548.

[35] Hongzuo Xu, Yijie Wang, Zhiyue Wu, and Yongjun Wang. 2019. MIX: A Joint Learning Framework for Detecting Both Clustered and Scattered Outliers in Mixed-Type Data. In *ICDM*. IEEE, 1408–1413.

[36] Xiao Zhang, Manish Marwah, I-ta Lee, Martin Arlitt, and Dan Goldwasser. 2019. ACE–An Anomaly Contribution Explainer for Cyber-Security Applications. In *International Conference on Big Data (Big Data)*. IEEE, 1991–2000.