Artificial Intelligence 339 (2025) 104275

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Out-of-distribution detection by regaining lost clues

Zhilin Zhao [a,b,*], Longbing Cao [a], Philip S. Yu [c,d]

[a] School of Computing, Macquarie University, Sydney, NSW 2109, Australia
[b] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China
[c] University of Illinois at Chicago, Chicago, IL, USA
[d] Institute for Data Science, Tsinghua University, Beijing, China

ARTICLE INFO

Keywords:
Out-of-distribution detection
Transformer
Ensemble learning
Generalization
AI safety

ABSTRACT

Out-of-distribution (OOD) detection identifies samples in the test phase that are drawn from distributions distinct from that of training in-distribution (ID) samples for a trained network. According to the information bottleneck, networks that classify tabular data tend to extract labeling information from features with strong associations to ground-truth labels, discarding less relevant labeling cues. This behavior leads to a predicament in which OOD samples with limited labeling information receive high-confidence predictions, rendering the network incapable of distinguishing between ID and OOD samples. Hence, exploring more labeling information from ID samples, which makes it harder for an OOD sample to obtain high-confidence predictions, can address this over-confidence issue on tabular data. Accordingly, we propose a novel transformer chain (TC), which comprises a sequence of dependent transformers that iteratively regain discarded labeling information and integrate all the labeling information to enhance OOD detection. The generalization bound theoretically reveals that TC can balance ID generalization and OOD detection capabilities. Experimental results demonstrate that TC significantly surpasses state-of-the-art methods for OOD detection in tabular data.

1. Introduction

Deep neural networks have demonstrated remarkable generalization capabilities in classifying tabular data. However, these networks operate on a fundamental assumption. They presume that training and test samples are independent and identically distributed (i.i.d.). These samples are drawn from the same unknown distribution, known as *In-Distributions* (ID). In practice, this assumption is often violated [1] because test samples may be associated with some distributions differing from that of training samples, i.e., *Out-Of-Distributions* (OOD) [2]. Network trained on ID data could make unexpected high-confidence predictions for OOD samples [3,4], which makes it challenging to distinguish between OOD and ID samples. This limitation poses a significant risk in real-world applications and underscores the need to enhance *OOD detection performance* [5] to ensure the safety of real-world AI systems [6].

From an information-theoretic perspective [7], a deep neural network learns to balance input compression with label prediction [8]. When classifying tabular data, the network prioritizes extracting information from the most relevant features for predicting the target label, which we define as *Strong Labeling Information*, while discarding less immediately relevant information, referred to as *Weak Labeling Information*. The features that contribute to strong labeling information are called *Strong Features*, as they directly influence the predictive accuracy. All other features, which are not as strongly associated with the target label, are classified as *Weak*

[*] Corresponding author at: School of Computing, Macquarie University, Sydney, NSW 2109, Australia.
E-mail address: zhilin.zhao@mq.edu.au (Z. Zhao).

https://doi.org/10.1016/j.artint.2024.104275
Received 19 November 2023; Received in revised form 13 November 2024; Accepted 10 December 2024

Available online 13 December 2024
0004-3702/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

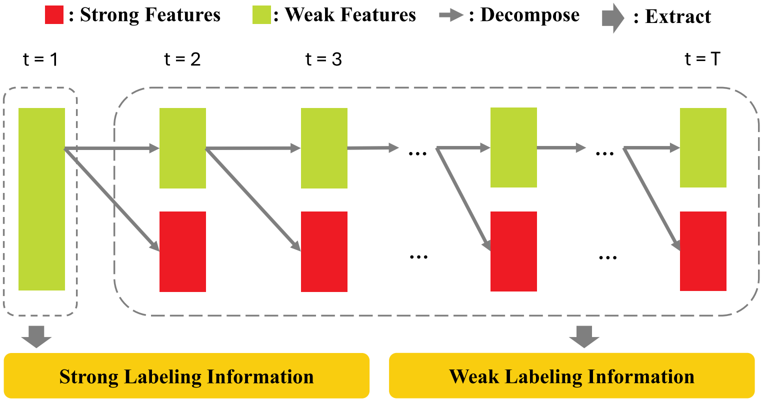


Fig. 1. Iterative Extraction of Weak Labeling Information. The first transformer treats all features as weak, adaptively selecting strong features and extracting strong labeling information while discarding some weak labeling information. Since a single transformer cannot fully capture all labeling information, an iterative approach is required. Each transformer processes the strong and weak features, then decomposes the weak features into new sets of strong and weak features for the next transformer to further explore the remaining weak labeling information. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Features. Weak labeling information comes from these weak features and from interactions between weak and strong features. While weak features may not be as predictive on their own, they offer valuable complementary information when considered alongside strong features. For example, in medical diagnosis, strong features like blood pressure are key predictors of heart disease. Weak features, such as diet or exercise, are less directly predictive but still provide useful context. By considering both, the network can make better predictions, especially in unusual cases.

Focusing solely on strong labeling information can lead to overconfidence when encountering OOD samples. Since the network primarily relies on the strong features, which are optimized for ID data, it may assign high confidence to OOD samples if these samples happen to align with the patterns learned from the strong features. This results in a failure to distinguish between ID and OOD samples, as the network lacks the nuanced information necessary to capture deviations from the learned ID patterns. In contrast, weak labeling information, derived from the weak features, plays a crucial role in OOD detection. Although these features are less directly related to label prediction, they provide complementary insights by capturing subtle relationships and interactions within the data that are often overlooked by focusing solely on strong features. By exploring these weak features, the network becomes more sensitive to deviations in feature distributions, which is a hallmark of OOD samples.

Based on the above analysis, when using a transformer for tabular data, a single transformer primarily focuses on strong features and discards some weak features for extracting strong labeling information. As a result, it overlooks the weak labeling information within the weak features and between the weak and strong features. This also implies that even if the discarded weak features are passed to another transformer, it still cannot fully extract all the weak labeling information. To address this, we propose the Transformer Chain (TC), illustrated in Fig. 1, which uses a sequence of dependent transformers to iteratively recover weak labeling information. The first transformer treats all features as weak and adaptively selects strong features, extracting strong labeling information. Each subsequent transformer processes the strong and weak features passed from the previous stage, exploring weak labeling information and further splitting weak features into new sets of strong and weak features. This iterative process ensures that weak labeling information is progressively captured, with all extracted information integrated to improve OOD detection.

The main contributions of this work include:

- The introduction of chained transformers, where the first transformer focuses on strong labeling information and subsequent transformers iteratively recover discarded weak labeling information, allows for the integration of both strong and weak labeling information to enhance robustness against OOD samples.
- A generalization error bound is derived to guide hyperparameter selection for balancing ID classification and OOD detection, with extensive experiments conducted to validate the effectiveness of the approach.

The remainder of this paper is structured as follows. Section 2 provides an overview of related techniques and research. Section 3 outlines the key concepts and definitions for the problem setup. Section 4 details the proposed TC method. Section 5 and Section 6 cover theoretical guarantees and present experimental results, respectively. Finally, Section 7 concludes this paper.

2. Related work

2.1. Transformer

Transformer, proposed for machine translation [9,10], has achieved significant progress in many natural language processing tasks. A sequence of words is fed into the standard transformer to characterize temporal relationships. Transformer-based models are usually pretrained on a large-scale unlabeled dataset using self-supervised learning [11] and fine-tuned on a small-scale dataset to adapt to a specific task. For example, Bidirectional Encoder Representations from Transformers (BERT) [12] pretrains deep bidirectional representations from unlabeled text according to a denoising self-supervised task. In addition, transformers have been applied to computer vision. Vision Transformer (ViT) [13] splits an image into several patches and feeds the patch sequence to a standard transformer encoder to capture spatial relationships. Masked autoencoders [14] built on ViT develop an asymmetric encoder-decoder architecture and adopt a high proportion of masking.

2.2. OOD detection

OOD Detection aims to distinguish OOD samples from ID samples during testing [15], which is closely related to outlier detection [16,17], where outliers are identified from the normal distribution during training [18]. However, while outlier detection focuses on identifying anomalous samples within the same distribution, OOD detection seeks to identify samples that come from entirely different distributions than those seen during training. Baseline methods like Maximum over Softmax Probabilities (MSP) [19] and Mahalanobis distance (MLB) [20] calculate OOD scores based on confidence and distance metrics, respectively. However, as post-hoc methods applied to pretrained networks, they fail to improve OOD robustness during training. More recent approaches modify training procedures to enhance robustness, such as Deep Gambler (DG) [21], Outlier Exposure (OE) [22], and Rectified Activations (RA) [23], but these are often designed for image data and overlook the complex feature interactions in tabular data. Additionally, deep Support Vector Data Description (DSVDD) and One-Class SVM extend outlier detection methods by learning hyperspheres around ID samples to detect anomalies [24]. These methods are effective for high-dimensional tabular data but still suffer from limited exploration of weak labeling information. In contrast, approaches like Deep Ensemble (DE) [25] combine multiple networks to enhance labeling information exploration. However, the random initialization strategy only partially addresses the insufficient exploration of weak labeling information, leaving room for more structured approaches that fully utilize both strong and weak labeling signals.

3. Preliminary

3.1. Definitions

We consider a training dataset $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ consisting of m i.i.d. samples from the joint distribution $\mathbb{P}$, each with n feature dimensions. Here, $\mathbf{x} \in \mathbb{R}^n$ denotes an input with n features (attributes) $\mathcal{F} = \{f_1, f_2, \dots, f_n \mid f_i \in [n], f_i \neq f_j \text{ for } i \neq j\}$, while $y \in [K]$ represents the corresponding ground truth label with K being the number of classes. Crucially, every feature of $\mathbf{x}$ is bounded, ensuring that $|\mathbf{x}(i)| \leq B$ for all $i \in [n]$. We assume $\mathcal{H}$ is the hypothesis space comprising transformers. Each hypothesis $h \in \mathcal{H}$ processes the input $\mathbf{x}$ with n^s strong features $\mathcal{F}^s \subseteq \mathcal{F}$ and n^w weak features $\mathcal{F}^w \subseteq \mathcal{F}$ $(\mathcal{F}^s \bigcap \mathcal{F}^w = \emptyset)$, mapping it to a representation containing labeling information. The loss function $|\mathcal{L}(h(\mathbf{x}, \mathcal{F}^s, \mathcal{F}^w), y)| \leq c$ for the hypothesis space $\mathcal{H}$ is bounded by a constant c and is L-Lipschitz continuous.

3.2. Information bottleneck

According to the information bottleneck principle [26], a network from the hypothesis space $\mathcal{H}$ restricts the amount of information extracted from the input space $\mathcal{X}$ to predict the labels in $\mathcal{Y}$. This principle reflects a trade-off between compressing the input and accurately predicting the labels, expressed as:

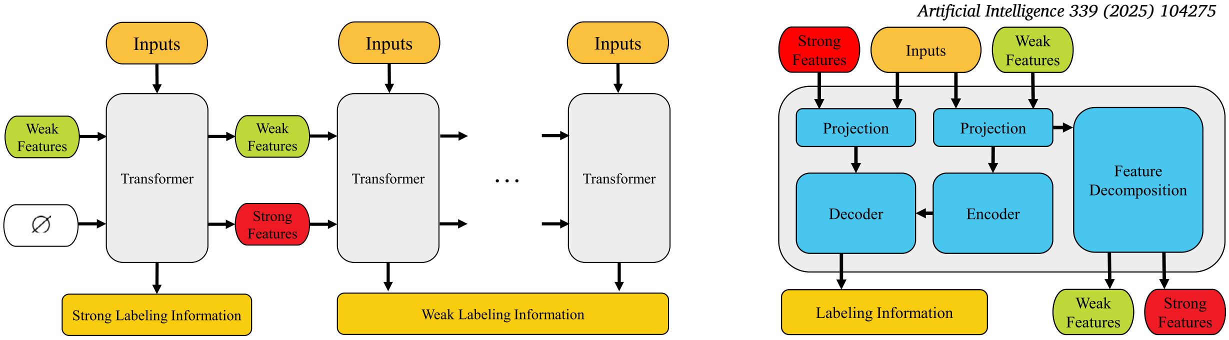
$$\max_{\mathcal{D}} \mathcal{I}(\mathcal{D};\mathcal{Y}) - \mathcal{I}(\mathcal{X};\mathcal{D}), \tag{1}$$

where $\mathcal{D}$ represents the intermediate feature representation learned by the network, and $\mathcal{I}(\cdot;\cdot)$ denotes the mutual information between two random variables, quantifying their mutual dependence. By selecting the most relevant strong features, the network compresses the input $\mathcal{X}$ to maximize $\mathcal{I}(\mathcal{D};\mathcal{Y})$, which measures the amount of strong labeling information captured in $\mathcal{D}$. In other words, $\mathcal{D}$ contains the strong labeling information necessary for accurate label prediction, while $\mathcal{I}(\mathcal{X};\mathcal{D})$ quantifies the amount of input information retained in $\mathcal{D}$.

However, $\mathcal{D}$ does not capture all the labeling information available in $\mathcal{X}$. This can be shown by decomposing the total mutual information between $\mathcal{X}$ and $\mathcal{Y}$ using the chain rule:

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{I}(\mathcal{D};\mathcal{Y}) + \mathcal{I}(\mathcal{C};\mathcal{Y} \mid \mathcal{D}), \tag{2}$$

where $\mathcal{I}(\mathcal{C};\mathcal{Y} \mid \mathcal{D})$ represents the conditional mutual information, which accounts for the additional labeling information in $\mathcal{C}$ that is not captured by $\mathcal{D}$. This reflects the weak labeling information. Since $\mathcal{I}(\mathcal{C};\mathcal{Y} \mid \mathcal{D}) > 0$ in most cases, it indicates that $\mathcal{D}$ alone is insufficient for capturing all labeling information. Thus, while $\mathcal{D}$ contains the strong labeling information, the weak labeling information is discarded. The weak labeling information is instead stored in $\mathcal{C}$, which captures subtler relationships between features that are not prioritized during the compression of $\mathcal{X}$ into $\mathcal{D}$. Therefore, the presence of $\mathcal{C}$ compensates for this loss, offering a more complete picture of the labeling information.



(a) Transformer Chain

(b) Transformer

Fig. 2. Model overview. The *input* refers to the raw feature values used for predictions, while *features* represent the indices used to select and process specific parts of the input. (a) The transformers in the chain iteratively explore the weak labeling information discarded by the first transformer. (b) A transformer in TC contains a projection module for selecting corresponding feature values and preforms embedding transformation, an encoder extracts weak labeling information within the weak features, a decoder extracts weak labeling information between the strong and weak features, and a feature decomposition module that splits the weak features into two parts for further processing by the next transformer. Note that the first transformer treats all features as weak and uses the encoder to adaptively select strong features from them, extracting the strong labeling information.

3.3. Implications for OOD detection

The information bottleneck principle shows that networks prioritize strong labeling information in $\mathcal{D}$ for label prediction, often discarding weak labeling information, which is crucial for OOD detection. OOD samples, differing from ID data, may still align with the strong features in $\mathcal{D}$, leading to overconfident predictions. However, $\mathcal{C}$, which contains weak labeling information, captures subtle patterns essential for distinguishing OOD samples. Incorporating this weak labeling information enhances the network sensitivity to deviations, improving its ability to detect OOD samples and addressing overconfidence issues, ultimately leading to greater robustness.

4. Transformer chain

As shown in Fig. 2a, the proposed Transformer Chain (TC) method utilizes a sequence of interconnected transformers to integrate both strong and weak labeling information, thereby enhancing OOD detection. The first transformer treats all features as weak and adaptively selects strong features from them, extracting strong labeling information and discarding the remaining weak features. Each subsequent transformer receives the strong and weak features from the previous transformer, exploring the weak labeling information both within the weak features and between the weak and strong features. This process iteratively decomposes the weak features into new sets of strong and weak features, enabling subsequent transformers to continue uncovering weak labeling information.

We assume TC contains T dependent transformers $\{h_t\}_{t\in[T]}$.[1] Each of these transformers $h_t(t \in [T])$ represents a hypothesis from the corresponding space $\mathcal{H}_t$ and receives n_t^s strong features $\mathcal{F}_t^s = \{f_{t,i}^s\}_{i\in[n_t^s]}$ and n_t^w weak features $\mathcal{F}_t^w = \{f_{t,i}^w\}_{i\in[n_t^w]}$ from the previous transformer to extract weak labeling information for learning representations. The weak feature set $\mathcal{F}_t^w$ is further split into a new set of strong features $\mathcal{F}_{t+1}^s = \{f_{t+1,i}^s\}_{i\in[n_{t+1}^s]}$ and weak features $\mathcal{F}_{t+1}^w = \{f_{t+1,i}^w\}_{i\in[n_{t+1}^w]}$. These two feature sets are then passed to the next transformer h_{t+1} for further extracting weak labeling information. It is worth noting that the first transformer h_1 receives all the features $\mathcal{F}$ without any prior knowledge of their strength. Therefore, all features can be considered as weak features, with no strong features present, i.e., $\mathcal{F}_1^s = \emptyset$ and $\mathcal{F}_1^w = \mathcal{F}$. It adaptively selects the strong features for the next transformer and extracts strong labeling information from them.

Specifically, as shown in Fig. 2b, for an input $\mathbf{x}$, each transformer h_t $(t \in [T])$ consists of a projection module $\mathbf{P}_t$ (Section 4.1) to select feature values and perform embedding transformation, an encoder $\mathbf{E}_t$ (Section 4.2) to extract weak labeling information within the weak features, a decoder $\mathbf{D}_t$ (Section 4.2) to extract weak labeling information between strong and weak features, and a feature decomposition module $\mathbf{F}_t$ (Section 4.3) that splits weak features for further processing by the next transformer. It is worth noting that since the first transformer does not receive strong features, its decoder is not functional. Instead, it uses only the encoder to adaptively select strong features from the entire feature set for extracting strong labeling information. The information extraction process can be expressed as:

$$h_t(\mathbf{x}, \mathcal{F}_t^s, \mathcal{F}_t^w) = \mathbf{D}_t\left(\mathbf{E}_t(\mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^w)), \mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^s)\right). \tag{3}$$

Furthermore, the feature decomposition process is expressed as:

$$\mathbf{F}_t(\mathcal{F}_t^w, h_t, q) = \{\mathcal{F}_{t+1}^s, \mathcal{F}_{t+1}^w\}, \tag{4}$$

[1] $[T]$ represents the set $\{1, 2, \ldots, T\}$, which is a shorthand for the integers from 1 to T.

where $q \in (0, 1)$ is a parameter governing the proportion of strong and weak features.

To integrate the labeling information from all transformers and enhance OOD detection, we apply an ensemble strategy based on deep ensemble learning [25]. The class probabilities from all transformers are combined to form the overall probability distribution for the input $\mathbf{x}$:

$$\mathbf{p}(\mathbf{x}) = \frac{1}{T} \sum_{t\in[T]} \mathrm{softmax}(h_t(\mathbf{x}, \mathcal{F}_t^s, \mathcal{F}_t^w)), \tag{5}$$

where $\mathbf{p}(\mathbf{x})$ represents the average class probabilities and $\mathrm{softmax}(\cdot)$ is the softmax function. To quantify the OOD likelihood of an input sample $\mathbf{x}$, we compute an OOD score $\mathrm{Score}(\mathbf{x})$ by taking the maximum confidence from the class probabilities in $\mathbf{p}(\mathbf{x})$:

$$\mathrm{Score}(\mathbf{x}) = \max_{k\in[K]} \mathbf{p}(\mathbf{x})_k. \tag{6}$$

ID samples are expected to have high OOD scores, while OOD samples should have low scores.

For clarity, the overall algorithm of the proposed TC method is summarized in Algorithm 1. Following this, the functions and structures of $\mathbf{P}_t$, $\mathbf{E}_t$, $\mathbf{D}_t$, and $\mathbf{F}_t$ are explained.

Algorithm 1 Transformer chain.

```
Input: training data $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^m$,
       maximum number of transformers $T$,
       component parameter $q$.
for $t = 1, \dots, T$ do
    Construct $h_t$: $h_t(\mathbf{x}, \mathcal{F}_t^s, \mathcal{F}_t^w) = \mathbf{D}_t\left(\mathbf{E}_t(\mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^w)), \mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^s)\right)$
    Optimize $h_t$: $h_t \in \arg\min_{h\in\mathcal{H}_t} \frac{1}{m} \sum_{(\mathbf{x},y)\in\mathcal{S}} \mathcal{L}(h(\mathbf{x}, \mathcal{F}_t^s, \mathcal{F}_t^w), y)$
    Decompose features for $h_{t+1}$: $\mathbf{F}_t(\mathcal{F}_t^w, h_t, q) = \{\mathcal{F}_{t+1}^s, \mathcal{F}_{t+1}^w\}$
end for
Output: a chain of transformers: $\{h_1, \dots, h_T\}$
```

4.1. *Projection module* P

Each transformer h_t receives n_t^s strong features $\mathcal{F}_t^s$ and n_t^w weak features $\mathcal{F}_t^w$ from the previous transformer h_{t-1}. Therefore, for a given input $\mathbf{x}$, the first step is to extract the corresponding feature values. The set of n_t^s strong features $\mathcal{F}_t^s$ and n_t^w weak features $\mathcal{F}_t^w$ are extracted as follows:

$$\mathbf{S}_t^s(\mathbf{x}) = \{\mathbf{x}(f_{t,i}^s)\}_{i\in[n_t^s]} \in \mathbb{R}^{n_t^s}, \quad \mathbf{S}_t^w(\mathbf{x}) = \{\mathbf{x}(f_{t,i}^w)\}_{i\in[n_t^w]} \in \mathbb{R}^{n_t^w}, \tag{7}$$

where $\mathbf{S}_t^s$ selects the values of the strong features from $\mathbf{x}$, and $\mathbf{S}_t^w$ selects the values of the weak features. This provides $\mathbf{x}_t^s$ as the set of selected strong feature values, and $\mathbf{x}_t^w$ as the set of selected weak feature values.

However, the resulting $\mathbf{x}_t^s$ and $\mathbf{x}_t^w$ cannot be directly applied to the transformer's encoder and decoder. This is because each feature needs to be represented as an embedding vector, rather than as a raw feature value. Additionally, as the number of input features increases, the memory requirements for processing them grow quadratically, making it inefficient for transformers to handle high-dimensional data directly. To address these challenges, we employ a strategy inspired by the word embedding technique used in natural language processing transformers. This approach maps all raw feature values to n_A high-level features, where each high-level feature is represented by an embedding of dimension n_M. Linear projections $\mathbf{W}_t^s \in \mathbb{R}^{(n_A\times n_M)\times n_t^s}$ and $\mathbf{W}_t^w \in \mathbb{R}^{(n_A\times n_M)\times n_t^w}$ are used to construct the projection module $\mathbf{P}_t$ for h_t:

$$\begin{aligned} \mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^s) &= \mathrm{Reshape}(\mathbf{W}_t^s \cdot \mathbf{S}_t^s(\mathbf{x}), (n_A, n_M)), \\ \mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^w) &= \mathrm{Reshape}(\mathbf{W}_t^w \cdot \mathbf{S}_t^w(\mathbf{x}), (n_A, n_M)). \end{aligned} \tag{8}$$

The *reshape* operation transforms the projected feature vector into a matrix of size (n_A, n_M), allowing the transformer to efficiently process the data.

4.2. *Encoder* E *and decoder* D

Each transformer h_t in the TC method contains an encoder $\mathbf{E}_t$ and a decoder $\mathbf{D}_t$. The combined effect of the encoder and decoder allows each transformer to systematically retrieve and process the weak labeling information that was previously overlooked, ensuring a more robust and comprehensive feature representation.

The encoder $\mathbf{E}_t$ processes the weak features $\mathcal{F}_t^w$ by applying self-attention layers to capture the weak labeling information within these features. This self-attention mechanism enables the encoder to explore dependencies and interactions among the weak features, which are often overlooked by conventional methods that focus primarily on strong features. By extracting subtle, often ignored information from weak features, the encoder enhances the network's sensitivity to feature variations, a key factor in detecting OOD samples.

Meanwhile, the decoder $\mathbf{D}_t$ is designed to capture the weak labeling information that exists between the strong and weak features. It takes the strong features $\mathcal{F}_t^s$ and the outputs from the encoder, using cross-attention layers to extract information from the interactions between the strong and weak features. This approach ensures the network fully leverages both feature sets, incorporating valuable insights from weak feature interactions, rather than relying solely on strong labeling information.

4.3. *Feature decomposition module* **F**

The transformer h_t applies the feature decomposition module $\mathbf{F}_t(\mathcal{F}_t^w, h_t, q)$ to decompose the received weak features $\mathcal{F}_t^w$ into $\mathcal{F}_{t+1}^s$ and $\mathcal{F}_{t+1}^w$, where $q \in (0,1)$ is a component parameter that controls the proportion of strong and weak features. We should consider the weights in the linear projection $\mathbf{W}_t^w$ because these weights determine how much emphasis is placed on each weak feature during the projection process. Specifically, the magnitude of the weights reflects the contribution of each weak feature to the overall representation. By analyzing the weights, we can estimate the relative importance of each weak feature. This allows us to selectively promote the most important weak features to strong features for the next iteration, ensuring that the model progressively focuses on the most informative features.

Accordingly, for each weak feature $f_i^w \in \mathcal{F}_t^w$, we calculate its importance by computing the L2-norm of the corresponding column in $\mathbf{W}_t^w$:

$$\text{Importance}(f_i^w) = \|[\mathbf{W}_t^w]_{:,:,i}\|_2 = \sqrt{\sum_{j_1}\sum_{j_2}([\mathbf{W}_t^w]_{j_1,j_2,i})^2} \tag{9}$$

Next, we sort the weak features $\mathcal{F}_t^w$ in descending order based on their importance. After sorting, the number of weak features n_{t+1}^w for the next step is defined as

$$n_{t+1}^w = \lfloor q^{t+1} n \rfloor, \quad n_{t+1}^s = q^t n - \lfloor q^{t+1} n \rfloor, \tag{10}$$

The top n_{t+1}^s features with the highest importance are selected as the new strong features $\mathcal{F}_{t+1}^s$, while the remaining n_{t+1}^w features are assigned to the new weak feature set $\mathcal{F}_{t+1}^w$. Thus, the new sets are defined as:

$$\mathcal{F}_{t+1}^s = \{f_{t+1,i}^s\}_{i \in [n_{t+1}^s]}, \quad \mathcal{F}_{t+1}^w = \{f_{t+1,i}^w\}_{i \in [n_{t+1}^w]}. \tag{11}$$

By splitting the weak features in this manner, we ensure that the most important weak features, as determined by the weights in $\mathbf{W}_t^w$, are promoted to strong features for further processing in the next transformer stage. Throughout the training process, the number of received features gradually decreases, indicating the number of transformers in the chain is finite. The maximum number of transformers is $\lfloor \log_{1/q} n \rfloor$. As a result, the number of transformers in TC is limited to $T \leq \lfloor \log_{1/q} n \rfloor$.

5. Theoretical guarantees

The component parameter $q \in (0,1)$ deciding the maximum number of transformers plays a pivotal role in TC, impacting both its effectiveness and efficiency. In this section, we present theoretical guarantees for TC, shedding light on how the hyper-parameter q influences classification generalization. By understanding these theoretical foundations, we can make informed decisions about selecting an appropriate value for q to achieve a balanced trade-off between various aspects of TC, such as ID classification generalization and OOD detection, as well as the trade-off between effectiveness and efficiency.

TC applies a uniformly-weighted mixture model to integrate the outputs of all transformers. Therefore, a hypothesis h of TC can be treated as a hypothesis from an ensemble hypothesis space $\mathcal{H}_{\text{conv}} = \text{conv}(\bigcup_{t=1}^T \mathcal{H}_t)$, i.e., $h \in \mathcal{H}_{\text{conv}}$. For a hypothesis $h \in \mathcal{H}_{\text{conv}}$ and a sample $(\mathbf{x}, y)$, we have,

$$h(\mathbf{x}) = \sum_{t \in [T]} \alpha_t h_t(\mathbf{x}, \mathcal{F}_t^s, \mathcal{F}_t^w), \tag{12}$$

where $\sum_{t \in [T]} \alpha_t = 1$. TC considers a subspace $\mathcal{H}_{\text{sub}}$ of $\mathcal{H}_{\text{conv}}$, i.e., $\mathcal{H}_{\text{sub}} \subseteq \mathcal{H}_{\text{conv}}$ where $\alpha_t = \frac{1}{T}$. The corresponding empirical and expected risks are given by

$$\epsilon_{\mathcal{S}}(h) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},y) \in \mathcal{S}} \mathcal{L}(h(\mathbf{x}), y), \quad \epsilon_{\mathbb{P}}(h) = \mathbb{E}_{(\mathbf{x},y) \in \mathbb{P}} \mathcal{L}(h(\mathbf{x}), y). \tag{13}$$

Accordingly, we define the minimizer $\hat{h} \in \arg\min \epsilon_{\mathcal{S}}(h)$ and the optimal solution $h^* \in \arg\min \epsilon_{\mathbb{P}}(h)$. The following theorem provides a generalization error bound for TC when the number of transformers is maximal, i.e., $T = \lfloor \log_{1/q} n \rfloor$.

We base our analysis on Rademacher complexity, which measures the capacity of a hypothesis class to fit random noise. This provides insights into the generalization performance of TC when the number of transformers is maximized. Rademacher complexity is formally defined as:

$$\mathcal{R}(\mathcal{H} \circ \mathcal{S}) = \frac{1}{|\mathcal{S}|} \mathbb{E}_{\sigma \sim \{\pm 1\}^{|\mathcal{S}|}} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{|\mathcal{S}|} \sigma_i h(\mathbf{x}_i) \right], \tag{14}$$

where $\mathcal{S}$ is the training set, σ_i are Rademacher random variables taking values in $\{-1, +1\}$, and $\sup_{h\in\mathcal{H}}$ represents the supremum over the hypothesis space $\mathcal{H}$. Analyzing the Rademacher complexity allows us to establish upper bounds on the generalization error of TC, showing how the hyper-parameter q impacts classification performance and generalization.

Theorem 1. *Assume the depth of each transformer is d and the Frobenius norm of the weight matrices are at most $M_1, \dots, M_d$. For any δ, with probability at least $1-\delta$, we have,*

$$\epsilon_{\mathbb{P}}(\widehat{h}) - \epsilon_{\mathbb{P}}(h^*) \le \mathcal{B}_1(q)\mathcal{B}_2(m, B, d, M_1, \dots, M_d) + \mathcal{B}_3(m, \delta),$$

where

$$\mathcal{B}_1(q) = \frac{1 - q^{\lfloor \log_{1/q} D \rfloor}}{(1-q)\lfloor \log_{1/q} D \rfloor},$$

$$\mathcal{B}_2(m, B, d, M_1, \dots, M_d) = \frac{2nB(\sqrt{2d\ln 2} + 1)\prod_{i=1}^{d} M_i}{\sqrt{m}},$$

$$\mathcal{B}_3(m, \delta) = 5c\sqrt{\frac{2\ln(8/\delta)}{m}}.$$

Proof. According to the Rademacher complexity [27] and Talagrand's contraction lemma [28], we have

$$\begin{aligned} &\epsilon_{\mathbb{P}}(\widehat{h}) - \epsilon_{\mathbb{P}}(h^*) \le 2\mathcal{R}(l \circ \mathcal{H}_{\text{sub}} \circ \mathcal{S}) + 5c\sqrt{\frac{2\ln(8/\delta)}{m}}, \\ &\mathcal{R}(l \circ \mathcal{H}_{\text{sub}} \circ \mathcal{S}) \le L\mathcal{R}(\mathcal{H}_{\text{sub}} \circ \mathcal{S}). \end{aligned} \tag{15}$$

According to the Jensen's inequality [29] and $h = \frac{1}{T}\sum_{t=1}^{T} h_t$, we have

$$\mathcal{R}(\mathcal{H}_{\text{sub}} \circ \mathcal{S}) \le \frac{1}{T}\sum_{t=1}^{T} \mathcal{R}(\mathcal{H}_t \circ \mathcal{S}). \tag{16}$$

According to the Rademacher bound for networks [30], for each hypothesis space $\mathcal{H}_t$, we have

$$\mathcal{R}(\mathcal{H}_t \circ \mathcal{S}) \le \frac{nB(\sqrt{2d\ln 2} + 1)\prod_{i=1}^{d} M_i}{\sqrt{m}}. \tag{17}$$

Recall that the input of h_t is $\mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^s) \oplus \mathbf{P}_t(\mathbf{x}, \mathcal{F}_t^w)$ where $\oplus$ denotes the concatenation operation. The input bound of h_1 is nB, and the input bound of $h_t, \forall (t \ge 2)$ is $q^{t-1}nB$. Accordingly, we have

$$nB + \sum_{t=2}^{T} q^{t-1} nB = B\frac{(1-q^{T-1}) + 1 - q}{1-q} \le 2B\frac{1-q^{T-1}}{1-q} \le 2B\frac{1-q^{T}}{1-q}, \tag{18}$$

where the two inequalities are due to $q \in (0, 1)$. Based on Eq. (18), we obtain the following bound by substituting Eq. (17) into Eq. (16),

$$\mathcal{R}(\mathcal{H}_{\text{sub}} \circ \mathcal{S}) \le \frac{2nB(\sqrt{2d\ln 2} + 1)(1-q^T)\prod_{i=1}^{d} M_i}{(1-q)T\sqrt{m}}, (T \ge 2). \tag{19}$$

We complete the proof by substituting Eq. (19) into Eq. (15). □

Lemma 1. *The function $g(x) = \frac{1 - x^{\log_{1/x} n}}{(1-x)\log_{1/x} n}$, where $x \in (0, 1)$, is a monotonically decreasing function for any $n > 1$.*

Proof. We start by evaluating $x^{\log_{1/x} n}$:

$$x^{\log_{1/x} n} = \left(\exp\left(\ln\left(\frac{1}{x}\right)\right)^{\frac{1}{\ln x}}\right)^{\ln n} = \frac{1}{e^{\ln n}}, \tag{20}$$

which simplifies to $\frac{1}{e^{\ln n}}$. Next, we calculate the term $(1-x)\log_{1/x} n$:

$$(1-x)\log_{1/x} n = (1-x)\frac{\ln n}{\ln \frac{1}{x}} = (x-1)\frac{\ln n}{\ln x}. \tag{21}$$

Now, we find the gradient of $\frac{x-1}{\ln x}$:

$$\left(\frac{x-1}{\ln x}\right)' = \frac{\ln x - \frac{1}{x}(x-1)}{\ln^2 x} = \frac{\ln x - 1 + \frac{1}{x}}{\ln^2 x}. \tag{22}$$

Since the gradient is positive, we conclude that $g(x)$ is a decreasing function for $x \in (0, 1)$. □

As per Theorem 1, the generalization error of TC can be attributed to three terms. $\mathcal{B}_3(m, \delta)$ is unavoidable due to the limited observed training ID samples from the unknown distribution $\mathbb{P}$. $\mathcal{B}_2(m, B, d, M_1, \dots, M_d)$ is influenced by the properties of transformers. Specifically, it suggests that a smaller Frobenius norm of weight matrices can lead to better generalization, highlighting the importance of network complexity in achieving significant performance. $\mathcal{B}_1(q)$ is linked to the characteristics of the proposed TC. According to Lemma 1, $\mathcal{B}_1(q)$ is a monotonic decreasing function w.r.t. q. Therefore, a larger $q \to 1$ leads to a better generalization because a larger q leads to a sharper generalization error bound $\epsilon_{\mathbb{P}}(\hat{h}) - \epsilon_{\mathbb{P}}(h^*)$. Furthermore, increasing q also enhances OOD detection performance, as it leads to more transformers being applied to explore additional weak labeling information. However, it is important to note that an excessive number of training transformers can significantly reduce efficiency. To strike a balance between ID classification and OOD detection performance, as well as effectiveness and efficiency, we generally recommend using $q = 0.5$. We will delve into the impact of the hyper-parameter q in greater detail in the section dedicated to hyper-parameter analysis.

6. Experimental results

We showcase the effectiveness of TC in both classifying ID samples and detecting OOD samples.[2]

6.1. Experimental setup

To verify the effectiveness of TC, we benchmarked it against several baseline and state-of-the-art methods, including Maximum over Softmax Probabilities (MSP) [19], MahaLanoBis distance (MLB) [20], Rectified Activations (RA) [23], Deep Support Vector Data Description (DSVDD) [24], Deep Gambler (DG) [21], Outlier Exposure (OE) [22], Minimum Other Score (MOS) [31], and Deep Ensemble (DE) [25]. MSP is a detector applied to a trained network without considering OOD sensitivity. The state-of-the-art methods improve upon the baseline MSP in various ways, such as enhancing the detection mechanism (MLB), introducing novel activation functions (RA), training an one-class classifier (DSVDD), optimizing the loss function (DG), refining the training process (OE), exploring different output spaces (MOS), and employing ensemble learning techniques (DE). For a fair comparison, all methods utilize the same transformer-based backbone architecture. Additionally, the ensemble versions of the four trainable methods, including DSVDD, DG, OE, and MOS, are used in the experiments, with the number of ensembles set to match the number of transformers in TC to ensure consistency across methods.

Unless otherwise specified, for the proposed TC method, the number of high-level features is set to $n_A = 100$, and the number of feature dimensions to $n_M = 128$. These configurations are considered sufficient for constructing a robust transformer for tabular data. If not specified, the component parameter is set to $q = 0.5$, and the number of transformers is determined as $T = \lfloor \log_{1/q} n \rfloor$. For a fair comparison, the number of transformers in DE is set to match that of TC, i.e., $\lfloor \log_{1/q} n \rfloor$ independent transformers are used in DE. While TC incorporates multiple networks and is conceptually related to ensemble learning, it differs from traditional ensemble methods in that the networks in TC are interdependent. TC focuses primarily on recovering discarded weak labeling information to enhance OOD detection performance.

Our evaluation follows the established practices of existing OOD detection methods [32]. Unless otherwise specified, we apply the baseline method MSP to calculate OOD scores for test samples. We then use the area under the receiver operating characteristic curve (AUROC) [33] and detection error to assess OOD detection performance. A higher AUROC value indicates a larger gap in OOD scores between ID and OOD samples, signifying better OOD detection performance. Conversely, a lower detection error reflects superior OOD detection. Additionally, we use accuracy to measure the ID classification performance.

6.2. OOD detection performance on tabular data

In order to validate the broad applicability of TC, we selected a diverse set of datasets. These datasets encompass a wide range of characteristics, including large-scale data, high-dimensional data, and data with a large number of labels. We chose eight datasets, five of which (Stellar, Skyserver, Gisette, SHABD, Speech) are sourced from Kaggle,[3] while the remaining three (Arrhythmia, Gene, Wine) are from the UCI machine learning repository.[4] The characteristics of these datasets are shown in Table 1. To construct ID and OOD samples for each dataset, we designate the samples from the smallest class as OOD and the samples from the other classes as ID. Subsequently, we partition the ID samples into training and test sets with an 8 : 2 ratio. In cases where the number of labels is two, an extra class is introduced during training to prevent all test samples from receiving 100% confidence.

[2] The source code is publicly available at: https://github.com/Lawliet-zzl/TC.

[3] https://www.kaggle.com/.

[4] https://archive.ics.uci.edu.

Table 1
Statistics of datasets.

Statistic	Stellar	Skyserver	Arrhythmia	Gisette	SHABD	Gene	Wine	Speech
# instances	100000	100000	87553	6000	243456	801	1143	3960
# features	16	17	187	5000	1024	20531	11	12
# labels	3	3	5	2	384	5	6	6

Table 2
OOD detection performance on tabular data. All values are in percentage, and the boldface values represent relatively better detection performance. Larger AUROC and lower Detection values indicate better performance. Results are averaged over five random trials.

	Dataset	MSP	MLS	RA	DSVDD	DG	OE	MOS	DE	TC
AUROC	Stellar	$63.7_{\pm 9.7}$	$54.3_{\pm 7.6}$	$61.4_{\pm 7.4}$	$64.6_{\pm 4.9}$	$74.9_{\pm 3.5}$	$64.9_{\pm 9.7}$	$66.7_{\pm 9.7}$	$66.1_{\pm 3.5}$	$\mathbf{68.0}_{\pm 5.1}$
	Skyserver	$90.0_{\pm 1.7}$	$78.8_{\pm 10.5}$	$86.2_{\pm 7.4}$	$82.3_{\pm 1.8}$	$79.1_{\pm 1.5}$	$91.2_{\pm 1.7}$	$91.6_{\pm 1.5}$	$91.0_{\pm 1.5}$	$\mathbf{94.0}_{\pm 2.0}$
	Arrhythmia	$70.4_{\pm 5.6}$	$62.1_{\pm 7.6}$	$71.5_{\pm 8.7}$	$65.3_{\pm 4.6}$	$71.6_{\pm 4.4}$	$73.6_{\pm 5.6}$	$74.1_{\pm 4.4}$	$73.3_{\pm 4.4}$	$\mathbf{75.9}_{\pm 1.6}$
	Gisette	$49.0_{\pm 2.9}$	$50.8_{\pm 3.0}$	$49.0_{\pm 2.9}$	$52.1_{\pm 4.8}$	$62.3_{\pm 3.2}$	$66.0_{\pm 2.9}$	$68.5_{\pm 3.2}$	$51.6_{\pm 3.2}$	$\mathbf{82.5}_{\pm 2.6}$
	SHABD	$94.8_{\pm 4.2}$	$49.5_{\pm 4.1}$	$94.3_{\pm 2.7}$	$70.8_{\pm 0.4}$	$96.1_{\pm 1.2}$	$95.6_{\pm 4.2}$	$97.7_{\pm 4.4}$	$96.2_{\pm 1.2}$	$\mathbf{99.5}_{\pm 0.6}$
	Gene	$54.0_{\pm 7.4}$	$53.0_{\pm 7.8}$	$54.0_{\pm 4.4}$	$52.0_{\pm 9.4}$	$61.5_{\pm 8.7}$	$85.0_{\pm 4.4}$	$81.4_{\pm 4.5}$	$73.5_{\pm 1.7}$	$\mathbf{99.1}_{\pm 2.6}$
	Wine	$50.3_{\pm 4.5}$	$47.4_{\pm 6.2}$	$50.3_{\pm 4.5}$	$51.3_{\pm 6.0}$	$52.6_{\pm 5.1}$	$56.3_{\pm 4.5}$	$55.2_{\pm 4.2}$	$51.2_{\pm 5.1}$	$\mathbf{58.8}_{\pm 8.4}$
	Speech	$50.7_{\pm 4.2}$	$49.8_{\pm 4.2}$	$50.7_{\pm 4.2}$	$50.3_{\pm 3.5}$	$49.8_{\pm 3.7}$	$51.9_{\pm 4.2}$	$51.2_{\pm 4.2}$	$50.7_{\pm 3.7}$	$\mathbf{52.9}_{\pm 1.7}$
Detection	Stellar	$41.7_{\pm 9.6}$	$43.0_{\pm 8.1}$	$33.8_{\pm 7.6}$	$38.6_{\pm 5.8}$	$34.4_{\pm 2.2}$	$31.2_{\pm 8.4}$	$32.0_{\pm 8.7}$	$33.8_{\pm 3.6}$	$\mathbf{30.4}_{\pm 5.3}$
	Skyserver	$17.2_{\pm 1.8}$	$22.5_{\pm 7.9}$	$14.3_{\pm 8.8}$	$30.2_{\pm 2.0}$	$20.7_{\pm 1.1}$	$13.9_{\pm 2.1}$	$13.8_{\pm 0.5}$	$12.8_{\pm 1.1}$	$\mathbf{12.6}_{\pm 2.4}$
	Arrhythmia	$35.3_{\pm 6.0}$	$34.0_{\pm 7.6}$	$31.5_{\pm 9.7}$	$32.7_{\pm 5.5}$	$31.5_{\pm 5.6}$	$28.8_{\pm 5.4}$	$28.8_{\pm 3.9}$	$28.2_{\pm 2.9}$	$\mathbf{28.5}_{\pm 1.4}$
	Gisette	$49.8_{\pm 3.8}$	$47.5_{\pm 3.0}$	$48.2_{\pm 3.2}$	$48.6_{\pm 6.8}$	$36.2_{\pm 3.1}$	$45.4_{\pm 1.9}$	$38.2_{\pm 3.9}$	$35.4_{\pm 2.3}$	$\mathbf{28.5}_{\pm 1.5}$
	SHABD	$15.7_{\pm 4.4}$	$25.8_{\pm 5.8}$	$14.7_{\pm 2.0}$	$14.6_{\pm 1.3}$	$11.0_{\pm 1.8}$	$11.2_{\pm 4.2}$	$11.2_{\pm 2.9}$	$11.3_{\pm 2.2}$	$\mathbf{9.3}_{\pm 2.7}$
	Gene	$48.9_{\pm 8.0}$	$34.7_{\pm 7.3}$	$32.2_{\pm 5.0}$	$31.4_{\pm 9.7}$	$33.9_{\pm 5.6}$	$32.0_{\pm 2.5}$	$31.1_{\pm 5.0}$	$33.5_{\pm 6.5}$	$\mathbf{24.7}_{\pm 3.2}$
	Wine	$45.4_{\pm 5.2}$	$43.9_{\pm 6.2}$	$43.7_{\pm 3.2}$	$41.3_{\pm 6.6}$	$37.2_{\pm 5.2}$	$42.8_{\pm 2.4}$	$40.9_{\pm 5.0}$	$42.4_{\pm 4.7}$	$\mathbf{35.0}_{\pm 7.9}$
	Speech	$49.4_{\pm 5.4}$	$45.6_{\pm 5.1}$	$47.1_{\pm 4.5}$	$48.5_{\pm 3.9}$	$44.4_{\pm 4.3}$	$43.1_{\pm 3.0}$	$43.7_{\pm 4.2}$	$46.0_{\pm 2.1}$	$\mathbf{42.6}_{\pm 0.3}$

The results of OOD sample detection on tabular data are presented in Table 2. TC outperforms MSP by a substantial margin, achieving a noteworthy improvement in AUROC ranging from 6.9% to 50.0%. Furthermore, when compared to all state-of-the-art methods, TC consistently records the highest AUROC scores across all datasets, indicating its superior sensitivity to OOD samples. This enhanced performance can be attributed to ability of TC to capture more labeling information during training, reducing the likelihood of OOD samples receiving high-confidence predictions by extracting minimal labeling information. Specifically, TC leverages the first transformer to explore strong labeling information, while subsequent transformers focus on weak labeling information. These two types of labeling information complement each other, contributing to effectiveness of TC. It is worth noting that DE is the closest competitor to TC in terms of OOD sample detection. The strong performance of DE can be attributed to its integration of multiple randomly-initialized transformers, which collectively explore more labeling information than a single transformer. However, it is important to highlight a key distinction: the transformers in DE are independent and cannot recover discarded weak labeling information. Additionally, the labeling information explored by DE tends to overlap due to the random initialization strategy. In contrast, each transformer in TC sequentially investigates weak labeling information discarded by its predecessor, leading to more comprehensive OOD detection.

While the multi-transformer architecture in TC improves scalability to some extent, specific challenges remain for larger datasets and real-time applications. The sequential structure of transformers in TC introduces cumulative computational overhead as the model grows, which can slow down processing on high-dimensional, large-scale datasets. Additionally, the memory demands, particularly from large embedding matrices and attention layers, may strain resources, potentially exceeding memory and latency limits in real-time scenarios. To address these issues, optimizations such as early-exit mechanisms to reduce the number of transformers once sufficient labeling information is captured, or parameter sharing across transformers to lower memory usage, could prove effective. Model compression techniques like pruning or quantization may also help reduce computational load while maintaining detection accuracy. Further research into these strategies could improve scalability and efficiency of TC, extending its applicability to real-time and large-scale environments.

6.3. OOD detection performance on image data

While TC is primarily designed for tabular data, exploring its performance on image data can strengthen the generalizability of the approach. This experiment, presented in Table 3, evaluates TC on image datasets, using CIFAR10 [34] as the ID dataset and SVHN [35], CIFAR100 [34], LSUN [36], and TinyImageNet [37] as the OOD datasets. TC demonstrates an average AUROC improvement of approximately 4.8% across the four OOD datasets.

However, there remains a performance gap when applying TC to image data due to fundamental differences between tabular and image data. In tabular data, each feature holds specific statistical significance, allowing feature selection mechanism of TC to iteratively explore weak labeling information effectively. This structured exploration aligns well with the characteristics of tabular data. Conversely, in image data, pixel-level features lack individual statistical significance, limiting the capacity of TC to leverage weak labeling information effectively. Thus, while TC shows promise in image data OOD detection, further refinements are essential

Table 3
OOD detection performance on image data. All values are in percentage, and the boldface values represent relatively better detection performance. Larger AUROC and lower Detection values indicate better performance.

Method	ID	SVHN	CIFAR100	LSUN	TinyImageNet
MSP		95.6	79.6	92.3	86.5
DE	CIFAR10	96.2	83.6	94.5	90.3
TC		**98.6**	**86.4**	**97.3**	**96.1**

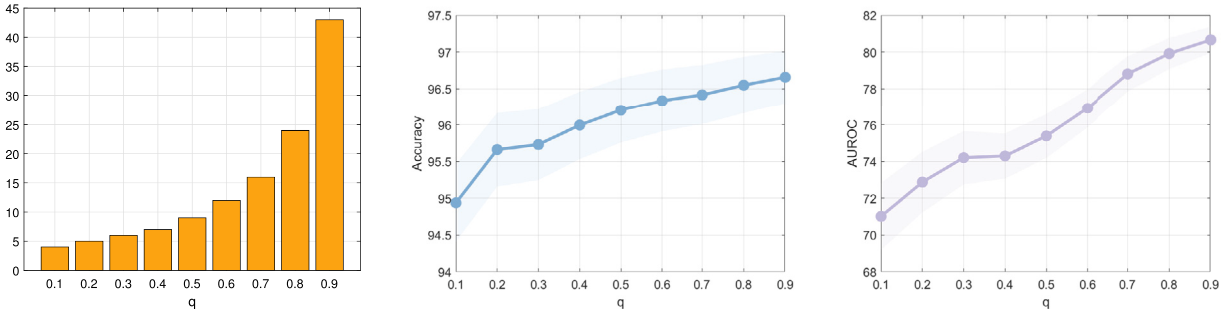


Fig. 3. Effect of the component parameter q on ID classification and OOD detection.

to fully exploit its potential in this domain. A more detailed analysis on how TC could adapt its feature exploration to high-dimensional image data may provide valuable insights for improving its applicability beyond tabular contexts.

6.4. Effect of the component parameter

To assess the impact of the component parameter q, which determines the maximum number of transformers in TC, we conducted an empirical analysis. We tested q with a set of nine evenly spaced values ranging from 0.1 to 0.9 on the Arrhythmia dataset. Increasing the value of q results in a greater number of transformers in the chain. This is consistent with the design principle, where a larger q assigns more weak features to each transformer, necessitating more transformers to fully explore the discarded weak labeling information. The effects of varying q on both ID classification accuracy and OOD detection performance (measured by AUROC) are presented in Fig. 3. As the figure shows, increasing the component parameter q leads to improvements in both ID classification accuracy and OOD detection performance. This empirical finding aligns with the theoretical results revealed by Theorem 1. The increase in q corresponds to more transformers being used to explore weak labeling information, enabling the network to acquire more knowledge for distinguishing ID samples and making it more challenging for OOD samples to receive high-confidence predictions. Notably, the most significant improvements in both accuracy and AUROC are observed when $q = 0.5$. It is important to mention that a larger q demands more computational resources due to the increased number of transformers. Therefore, to strike a balance between ID classification and OOD detection, as well as effectiveness and efficiency, we recommend adopting $q = 0.5$ for TC, unless specific considerations dictate otherwise.

6.4.1. Effect of the hyperparameters in linear projections

To assess the impact of the number of high-level features n_A and the dimension of feature embeddings n_M in the linear projections W^s and W^w, which influence the input scale of the encoder and decoder components in each transformer within the chain, we conducted experiments with various values. Specifically, we considered n_A values from the set $\{25, 100, 150\}$ and n_M values from $\{16, 32, 64, 128, 256, 512\}$. The experimental results, measured in terms of accuracy and AUROC, are presented in Fig. 4. Our findings indicate that TC exhibits robustness with respect to these two hyperparameters. It is important to note that these parameters determine the scale of the weight matrices in the linear projections. Based on our experimental results and observations, we have selected $n_A = 100$ and $n_M = 128$ as the values for constructing a powerful transformer tailored to tabular data. These settings strike a balance between computational efficiency and the network ability to capture both strong and weak labeling information.

6.5. Ablation study: TC vs. DE

We conducted a set of ablation experiments on the Arrhythmia dataset to compare the performance of TC with DE. Both methods employ multiple transformers, but they differ in how they utilize them. Specifically, TC uses chained and dependent transformers, whereas DE uses independently initialized transformers to randomly explore labeling information. In TC, each transformer decomposes the received weak features into strong and weak features, allowing the next transformer to explore the discarded weak labeling information. This weak labeling information complements the strong labeling information explored by the first transformer in both DE and TC. To verify that dependent transformers in TC can better explore weak labeling information, we compared TC with DE by varying the maximum number of transformers $T \in [1, \ldots, \lfloor \log_{1/q} n \rfloor]$.

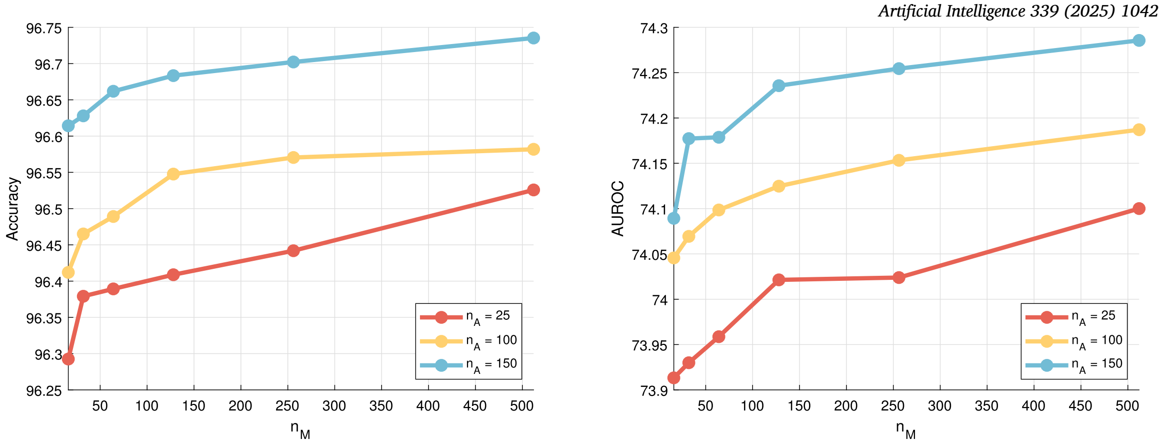


Fig. 4. Effect of the number of high-level features n_A and the dimension of feature embeddings n_M on ID classification and OOD detection.

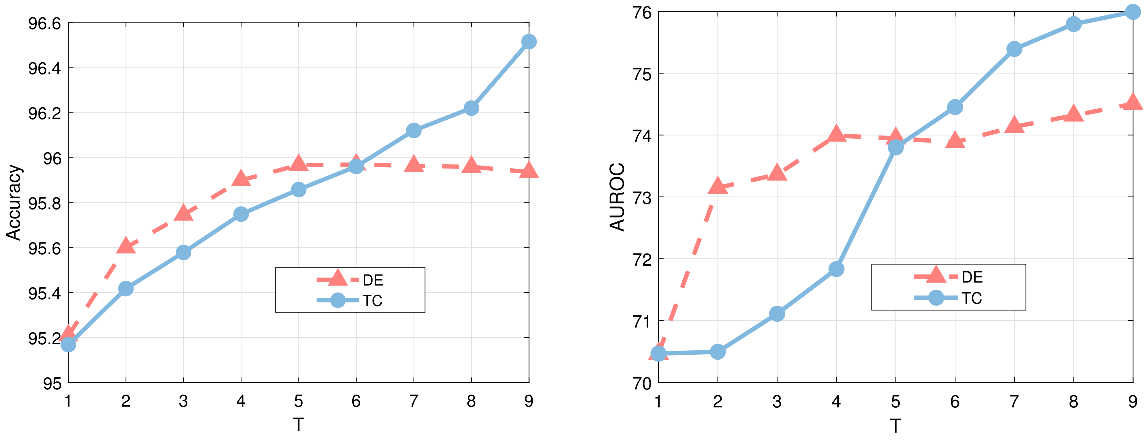


Fig. 5. Ablation Study. Both TC and DE employ multiple transformers, but they differ in how they process labeling information. TC uses chaining and dependency, while DE relies on independent random exploration.

As illustrated in Fig. 5, increasing the number of transformers T consistently improves both ID classification and OOD detection for TC and DE. However, the performance improvement of DE plateaus faster than that of TC, with DE showing an advantage only at lower values of T (up to $T \leq 5$). This difference arises because DE treats each transformer independently, with all transformers receiving the full set of input features. This independence results in redundant computation as each transformer processes the same feature dimensions. In contrast, the dependent transformer chain of TC reduces computational overhead by successively decreasing the number of features each transformer receives. As transformers in TC discard weak features and focus on the remaining information, the model becomes increasingly efficient as T increases, using fewer input dimensions while achieving higher accuracy.

Thus, while DE benefits from redundancy at smaller values of T, its performance diminishes at larger values due to inefficiencies in random exploration. TC, on the other hand, excels with a larger number of transformers by systematically leveraging dependencies across transformers, resulting in more efficient use of computational resources and more comprehensive OOD detection. This structured approach not only enhances the accuracy and sensitivity of TC but also reduces memory usage and processing time, making it better suited for high-dimensional datasets and resource-constrained applications.

However, a systematic ablation of each component is challenging due to the integrated nature of architecture of TC. Unlike traditional modular systems where each component can be isolated, the TC framework leverages the interdependence of its components to achieve its performance gains. For instance, the projection and decomposition modules dynamically adjust the features processed by each transformer in the chain, leading to a reduction in feature dimensionality as T increases, which is crucial for efficiency and scalability of TC. Without this progressive reduction, the model would incur significant computational overhead similar to that of independently initialized transformers in DE. Furthermore, removing components such as the encoder or decoder disrupts the feature selection process and weak labeling information extraction, making it impossible to achieve meaningful results without the full chain of components working in tandem. The role of each component is tightly woven into the iterative feature decomposition and labeling exploration process, which is what allows TC to progressively capture both strong and weak labeling information. This unique structure highlights the necessity of each component in maintaining the OOD detection capability and efficiency of TC, precluding simple component-wise ablation.

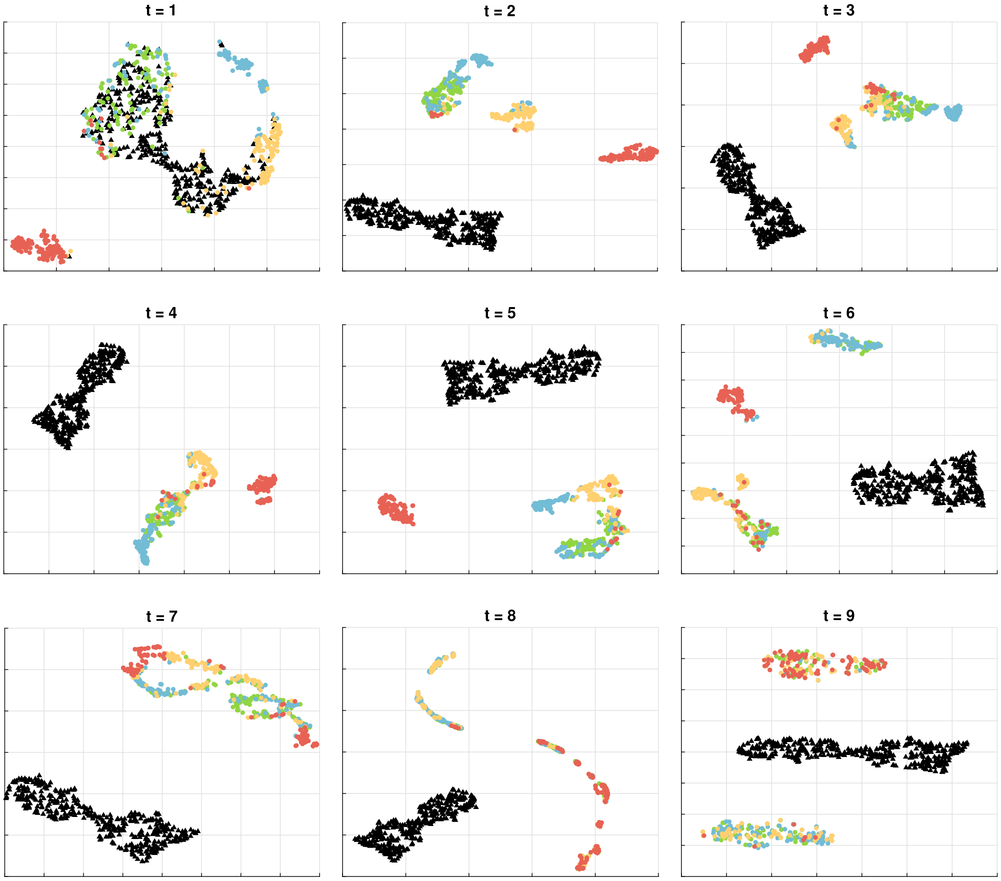


Fig. 6. Visualization by t-SNE of different transformers in the chain. The black triangle symbols represent OOD samples, and the other colored dots represent ID samples.

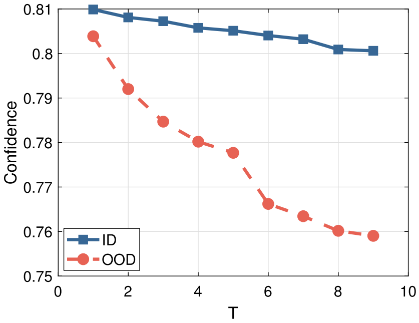


Fig. 7. Confidence between ID and OOD samples.

6.6. t-SNE visualizations and confidence gap

To gain deeper insights into the network performance in OOD detection, we construct t-SNE embeddings of the output latent representations from the second-to-last layer of transformers, as shown in Fig. 6. Additionally, we present the confidence gap between ID and OOD samples increases as the number of transformers grows, as shown in Fig. 7.

The t-SNE visualization in Fig. 6 highlights the distinct capabilities of each transformer in the chain to differentiate between ID and OOD samples. The first transformer ($t = 1$) primarily captures strong labeling information, leading to a clear distinction between ID and OOD samples. As we move through the chain, subsequent transformers ($t \geq 2$) continue to separate ID and OOD data, but they do so by exploring different weak labeling information. Each transformer uncovers unique features that contribute to OOD detection, highlighting that while the ID classification boundaries may shift or change, the ability to distinguish OOD data remains consistent across the chain. This variation in feature emphasis across transformers underscores the ensemble learning principle, where integrating the outputs of all transformers allows TC to leverage their complementary strengths, improving overall OOD detection.

The results in Fig. 7 show that as the number of transformers increases, the confidence gap between ID and OOD samples widens. This increasing gap allows the network to more effectively distinguish OOD samples, thereby improving OOD detection. The reason behind this improvement lies in the network progressive exploration of weak labeling information. As more transformers are added, weak labeling information becomes increasingly relevant in the decision-making process. For ID samples, this additional labeling information reinforces the confidence in correctly classifying them, while for OOD samples, the lack of strong label-related features makes it harder for them to achieve high confidence. This is because the network requires stronger label-related evidence to classify a sample as ID. As a result, OOD samples fail to meet these stricter criteria, leading to a significant drop in their confidence scores, which ultimately enhances the network ability to separate OOD from ID data.

7. Conclusion

A single transformer, primarily designed for strong labeling information, often assigns high-confidence predictions to OOD samples, even when they contain limited labeling information. To address this issue, we introduced a novel approach called the *transformer chain* (TC), specifically tailored to enhance OOD sensitivity in tabular data. TC comprises a sequence of transformers, with each one building upon the work of its predecessor to recover discarded weak labeling information. These transformers collectively capture both strong and weak labeling information to improve ID classification and OOD detection. To provide a solid theoretical foundation, we derived a generalization bound by analyzing the Rademacher complexity. These insights guide the selection of optimal hyperparameters, balancing ID classification generalization with OOD detection, as well as efficiency and effectiveness. Our empirical results demonstrate that TC consistently outperforms state-of-the-art methods in detecting OOD samples in tabular data.

Future work could extend this approach to image data, which presents unique challenges due to the lack of clearly defined, statistically significant features as seen in tabular data. Applying TC to image data would require further refinement in how weak labeling information is captured and explored within high-dimensional, pixel-based features.

CRediT authorship contribution statement

Zhilin Zhao: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Longbing Cao:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization. **Philip S. Yu:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was partially supported by the Australian Research Council (ARC) through Linkage Grant LP230201022, Discovery Grant DP240102050, and Linkage Infrastructure, Equipment and Facilities Grant LE240100131.

Data availability

Data will be made available on request.

References

[1] L. Cao, P.S. Yu, Z. Zhao, Shallow and deep non-iid learning on complex data, in: SIGKDD, 2022, pp. 4774–4775.
[2] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M.H. Rohban, M. Sabokrou, A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: solutions and future challenges, in: CoRR, 2021, pp. 1–38.
[3] Z. Zhao, L. Cao, K. Lin, Revealing the distributional vulnerability of discriminators by implicit generators, IEEE Trans. Pattern Anal. Mach. Intell. 45 (7) (2023) 8888–8901.
[4] Z. Zhao, L. Cao, K. Lin, Out-of-distribution detection by cross-class vicinity distribution of in-distribution data, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–12.
[5] W. Liu, X. Wang, J.D. Owens, Y. Li, Energy-based out-of-distribution detection, in: NeurIPS, 2020, pp. 1–12.
[6] D. Amodei, C. Olah, J. Steinhardt, P.F. Christiano, J. Schulman, D. Mané, Concrete problems in AI safety, CoRR (2016) 1–29.
[7] A.M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B.D. Tracey, D.D. Cox, On the information bottleneck theory of deep learning, in: ICLR, 2018, pp. 1–27.
[8] Z. Zhao, L. Cao, Dual representation learning for out-of-distribution detection, Trans. Mach. Learn. Res. (2023) 1–21.
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.
[10] K. Lin, J. Zhou, W. Zheng, Human-centric transformer for domain adaptive action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2024) 1–18.
[11] S. Mohseni, M. Pitale, J.B.S. Yadawa, Z. Wang, Self-supervised learning for generalizable out-of-distribution detection, in: AAAI, 2020, pp. 5216–5223.
[12] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: transformers for image recognition at scale, in: ICLR, 2021, pp. 1–22.

[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R.B. Girshick, Masked autoencoders are scalable vision learners, in: CoRR, 2021, pp. 1–14.

[15] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J.F. Núñez, J. Luque, Input complexity and out-of-distribution detection with likelihood-based generative models, in: ICLR, 2020, pp. 1–15.

[16] Y. Almardeny, N. Boujnah, F. Cleary, A novel outlier detection method for multivariate data, IEEE Trans. Knowl. Data Eng. 34 (9) (2022) 4052–4062.

[17] K. Lin, J. Du, Y. Gao, J. Zhou, W. Zheng, Diversifying spatial-temporal perception for video domain generalization, in: NeurIPS, 2023, pp. 1–15.

[18] F. Angiulli, R. Ben-Eliyahu-Zohary, L. Palopoli, Outlier detection for simple default theories, Artif. Intell. 174 (15) (2010) 1247–1253.

[19] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in: ICLR, 2017, pp. 1–12.

[20] K. Lee, K. Lee, H. Lee, J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, in: NeurIPS, 2018, pp. 7167–7177.

[21] Z. Liu, Z. Wang, P.P. Liang, R. Salakhutdinov, L. Morency, M. Ueda, Deep gamblers: learning to abstain with portfolio theory, in: NeurIPS, 2019, pp. 10622–10632.

[22] D. Hendrycks, M. Mazeika, T.G. Dietterich, Deep anomaly detection with outlier exposure, in: ICLR, 2019, pp. 1–18.

[23] Y. Sun, C. Guo, Y. Li, ReAct: out-of-distribution detection with rectified activations, in: NeurIPS, 2021, pp. 144–157.

[24] L. Ruff, N. Görnitz, L. Deecke, S.A. Siddiqui, R.A. Vandermeulen, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: ICML, 2018, pp. 4390–4399.

[25] I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett, Simple and scalable predictive uncertainty estimation using deep ensembles, in: NeurIPS, 2017, pp. 6402–6413.

[26] M. Tschannen, J. Djolonga, P.K. Rubenstein, S. Gelly, M. Lucic, On mutual information maximization for representation learning, in: ICLR, 2020, pp. 1–16.

[27] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning from Theory to Algorithms, Cambridge University Press, 2014.

[28] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT Press, 2018.

[29] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[30] N. Golowich, A. Rakhlin, O. Shamir, Size-independent sample complexity, in: COLT, vol. 75, 2018, pp. 297–299.

[31] R. Huang, Y. Li, MOS: towards scaling out-of-distribution detection for large semantic space, in: CVPR, 2021, pp. 8710–8719.

[32] G. Shalev, Y. Adi, J. Keshet, Out-of-distribution detection using multiple semantic label representations, in: NeurIPS, 2018, pp. 7386–7396.

[33] S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, in: ICLR, 2018, pp. 1–27.

[34] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. Rep., 2009.

[35] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

[36] F. Yu, Y. Zhang, S. Song, A. Seff, J. Xiao, LSUN: construction of a large-scale image dataset using deep learning with humans in the loop, CoRR, arXiv:1506.03365, 2015.

[37] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, in: CVPR, 2009, pp. 248–255.