ICAD-LLM: One-for-All Anomaly Detection via In-Context Learning with Large Language Models

Zhongyuan Wu[1, 3], **Jingyuan Wang**[1, 2, 3, 4*], **Zexuan Cheng**[1, 3], **Yilong Zhou**[1, 3], **Weizhi Wang**[1, 3], **Juhua Pu**[1, 3], **Chao Li**[1, 3], **Changqing Ma**[5]

[1]School of Computer Science and Engineering, Beihang University, Beijing, China
[2] School of Economics and Management, Beihang University, Beijing, China
[3] MOE Engineering Research Center of Advanced Computer Application Technology, Beihang University, China
[4] MIIT Key Laboratory of Data Intelligence and Management, Beihang University, Beijing, China
[5]Capinfo Co., Ltd.

Abstract

Anomaly detection (AD) is a fundamental task of critical importance across numerous domains. Current systems increasingly operate in rapidly evolving environments that generate diverse yet interconnected data modalities—such as time series, system logs, and tabular records—as exemplified by modern IT systems. Effective AD methods in such environments must therefore possess two critical capabilities: (1) the ability to handle heterogeneous data formats within a unified framework, allowing the model to process and detect multiple modalities in a consistent manner during anomalous events; (2) a strong generalization ability to quickly adapt to new scenarios without extensive retraining. However, most existing methods fall short of these requirements, as they typically focus on single modalities and lack the flexibility to generalize across domains. To address this gap, we introduce a novel paradigm: In-Context Anomaly Detection (ICAD), where anomalies are defined by their dissimilarity to a relevant reference set of normal samples. Under this paradigm, we propose ICAD-LLM, a unified AD framework leveraging Large Language Models' in-context learning abilities to process heterogeneous data within a single model. Extensive experiments demonstrate that ICAD-LLM achieves competitive performance with task-specific AD methods and exhibits strong generalization to previously unseen tasks, which substantially reduces deployment costs and enables rapid adaptation to new environments. To the best of our knowledge, ICAD-LLM is the first model capable of handling anomaly detection tasks across diverse domains and modalities. The extended version of this paper will be available at https://github.com/nobody384/ICAD-LLM

Introduction

Anomaly detection (AD) is a critical task with pervasive importance across numerous domains.

However, most existing AD methods are primarily designed for single data modalities, and their ability to generalize to new, unseen scenarios is often limited (Yao et al. 2024). This creates a significant gap between academic research and the demands of real-world applications. For instance, in modern IT systems like e-commerce platforms, a single fault such as a payment failure can manifest concurrently as CPU spikes (time series), error logs (log data), and abnormal transaction records (tabular data). This situation calls for a universal model that can handle various data types effectively, thereby reducing the need for multiple, disparate solutions. Furthermore, as these business systems rapidly evolve with new services and architectures, existing AD models must adapt quickly to novel scenarios without complete retraining for fast deployment. Under these conditions, conventional approaches that require separate model training for each task or modality become operationally infeasible.

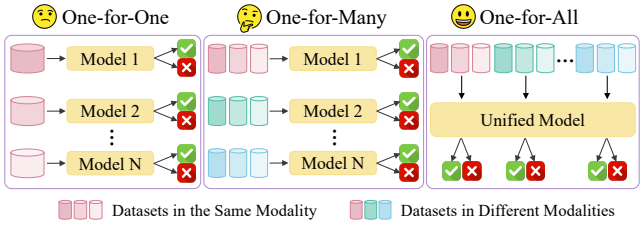

Figure 1: An illustration of different levels of AD.

In terms of their model-to-task mapping, as shown in Figure 1, current AD methods can be categorized into different levels. Traditional One-for-One AD (OFO-AD) methods train dedicated models for each specific dataset, learning the unique distribution of "normal" instances within that particular task (Li et al. 2003; Xu et al. 2018; Guo, Yuan, and Wu 2021; Yang et al. 2023; Yin et al. 2024). While effective for their designated tasks, these methods fail to generalize across different tasks or domains, necessitating costly retraining for each new application scenario. More recently, One-for-Many AD (OFM-AD) approaches have emerged as a response to the limitations of OFO-AD, improving generalization by enabling a single model to detect anomalies across various predefined tasks within the same modality (You et al. 2022; He et al. 2024; Yao et al. 2024; Li et al. 2023). However, these approaches still fall short of the requirements outlined above: they remain confined to that sin-

*Corresponding author. Email: jywang@buaa.edu.cn

gle data modality and lack the architectural versatility to be applied to other data types.

The aforementioned limitations of existing AD methods lead to a natural yet ambitious question: Is it possible to develop a "One-for-All" model capable of handling diverse tasks across multiple data modalities? To answer this question, we must revisit the original definition of anomalies by Grubbs (Grubbs 1969), which described anomalies as "*one that appears to deviate markedly from other members of the sample in which it occurs*". This principle, however, is contradicted by the design of most existing AD methods. By evaluating each sample individually, these methods prevent the model from directly comparing it to "other members" at inference time, which forces the model to rely on an internalized, static understanding of normality learned from the training set. This act of memorization binds the model to a specific task, restricting it from generalizing across tasks and modalities. This insight leads us to a new perspective: what if we could empower the AD model with the fundamental skill of comparison by providing the necessary context on-the-fly, thereby decoupling AD from task-specific distribution learning? To realize this vision, we introduce In-Context Anomaly Detection (ICAD). ICAD explicitly provides a reference set of normal samples during inference, and then assesses anomalies by comparing the target sample to this contextually relevant reference set. By shifting the objective from memorization to in-context comparison, this approach is inherently more flexible and readily applicable across diverse data modalities.

However, translating this high-level ICAD paradigm into a practical and effective One-for-All model is non-trivial. It necessitates a framework design that satisfies three key requirements (REQ): ***REQ1****-Feature Alignment*. Given that data from different modalities have vastly different feature dimensions and semantic structures, the model must first project these disparate inputs into a common embedding space, which is the foundational step that enables a single, unified architecture to process them meaningfully. ***REQ2****-Discrepancy-Sensitive Representation*. The model must extract rich, semantic representations that are not only modality-agnostic but also sensitive to the subtle dissimilarities between a target sample and its reference sets. ***REQ3****-Task-Agnostic Discriminative Objective*. Unlike traditional AD training objectives (e.g., minimizing reconstruction loss) that are tightly coupled to specific data distributions, ICAD requires a new training objective. This objective must decouple the model from the training data by explicitly training its universal ability to discriminate a target's dissimilarity against its reference set, rather than encouraging task-specific memorization.

To fulfill these requirements, we propose ICAD-LLM, a unified AD framework leveraging Large Language Models' in-context learning abilities to process heterogeneous data within a single model. ICAD-LLM consists of three key components, each tailored to satisfy a specific requirement. To meet ***REQ1***, we design a *Modality-Aware Encoder* that projects heterogeneous inputs from time-series, logs, and tables into a unified, fixed-dimension embedding space. To solve ***REQ2***, we employ a *Prompt-Guided Representation Module*. This component harnesses the in-context learning capacity of Large Language Models to extract semantically rich representations. To address ***REQ3***, we formulate a *Contextual Contrastive Learning* objective, which explicitly trains the model to discern subtle differences between normal and anomalous patterns. Crucially, ICAD-LLM is only trained once to acquire a general-purpose anomaly discrimination capability. At inference time, this single model can tackle anomaly detection tasks across diverse modalities without task-specific retraining. This "train-once, apply-broadly" strategy equips ICAD-LLM with much flexibility and efficiency. Extensive experiments show that ICAD-LLM achieves performance competitive with state-of-the-art task-specific methods and exhibits strong generalization to previously unseen tasks. To the best of our knowledge, ICAD-LLM is the first model capable of handling AD tasks across diverse domains and modalities. The main contributions of this paper are summarized as follows:

- We introduce In-Context Anomaly Detection, which redefines anomaly detection based on the concept of contextual dissimilarity, enabling a more generalized and flexible anomaly detection approach.
- We propose ICAD-LLM, a novel model designed to effectively implement the ICAD paradigm across multiple data modalities and diverse tasks.
- We demonstrate the ICAD-LLM achieves competitive performance on standard AD benchmarks and, more importantly, exhibits strong generalization to out-of-domain datasets without task-specific retraining.

Related Work

One-for-One Anomaly Detection

Traditional anomaly detection operates largely on a one-for-one paradigm, where a model is trained to fit the normal data of a single, specific dataset. This paradigm first flourished with classic machine learning methods (Breunig et al. 2000; Li et al. 2003; Liu, Ting, and Zhou 2008). In recent years, deep learning has advanced this paradigm for various data types by modeling it as a boundary-based task, which learns a compact feature hypersphere to enclose normal data. This is exemplified by a wealth of specialized models, such as OmniAnomaly (Su et al. 2019a), AnomalyTransformer (Xu et al. 2022) and DCdetector (Yang et al. 2023) in time series analysis; MCM (Yin et al. 2024) in tabular data; DeepLog (Du et al. 2017), LogAnomaly (Meng et al. 2019) and LogBert (Guo, Yuan, and Wu 2021) in system logs. However, their specialization in a single data distribution results in performance degradation when encountering new tasks.

One-for-Many Anomaly Detection

The emerging One-for-Many paradigm addresses the aforementioned generalization challenge by aiming to construct a single, unified model capable of processing multiple datasets, typically within the same modality. This progress follows two main technical routes. One is the design of novel, unified architectures, such as the memory-bank-based PatchCore (Roth et al. 2022), the feature-adaptation-based

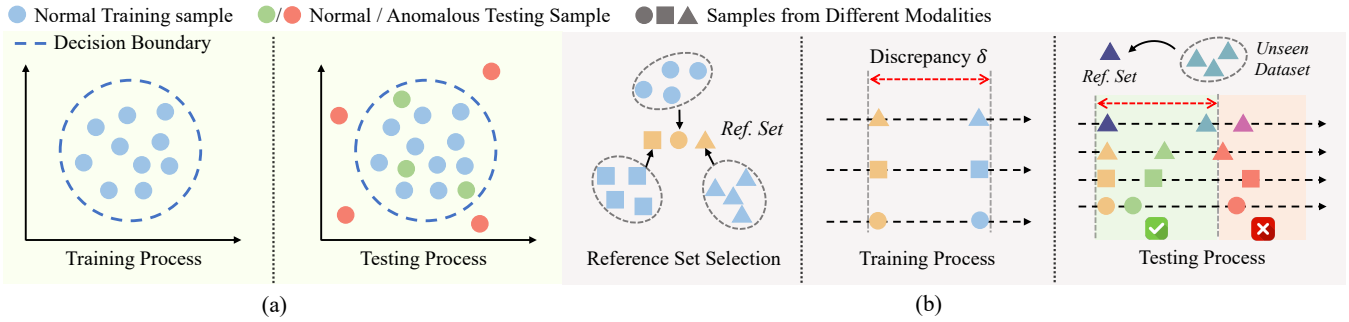

Figure 2: Comparison of different AD paradigms. (a) Traditional One-for-One/Many AD learns a fixed decision boundary from training data to identify outliers. (b) In-Context Anomaly Detection learns a general comparison function, detecting anomalies based on their discrepancy from a given reference set.

SimpleNet (Liu et al. 2023), and more recent frameworks like GOAD (Bergman and Hoshen 2020), UniAD (You et al. 2022), and MambaAD (He et al. 2024). The other route adapts large-scale, pre-trained models for this task, as demonstrated by PMAD (Yao et al. 2023), WinCLIP (Jeong et al. 2023), ResAD (Yao et al. 2024), and AnomalyLLM (Liu et al. 2024). Notably, PMAD was the first to explicitly advocate the "one-for-all" concept, aiming to deploy a single detector across different datasets without task-specific retraining. However, a critical limitation underlies these advancements: they still remain fundamentally modality-specific. Although some initial explorations have been made to bridge this cross-modal gap (Li et al. 2023), they still require training on each modality independently, thus falling short of a truly universal solution.

Preliminary

Let $\mathbb{M} = \{\mathcal{M}_i\}_{i=1}^{m}$ be the set of all possible m modalities and $\mathbb{T} = \bigcup_{i=1}^{m} T_{\mathcal{M}_i}$ denote the universe of tasks, where $T_{\mathcal{M}_i} = \{\tau_j^{(i)}\}_{j=1}^{n_i}$ represents the set of tasks under modality $\mathcal{M}_i$. For any task $\tau \in \mathbb{T}$, we define the task-specific data as $\mathcal{D}_\tau = \mathcal{X}_\tau \times \mathcal{Y}_\tau$, where $\mathcal{X}_\tau$ presents all the samples in task τ, and $\mathcal{Y}_\tau = \{0, 1\}$ denotes the binary labels indicating normal (0) versus anomalous (1) classes. For clarity, we denote the general anomaly detection process as $\mathcal{A}$. Figure 2 provides a visual comparison of different AD paradigms.

One-for-One/Many AD

Given a target sample $x_{tgt} \in \mathcal{X}_\tau$, both OFO-AD and OFM-AD aim to learn a score function f, formalizing the AD process as:

$$\mathcal{A}(x_{tgt}; f, \theta_\tau) = \mathbb{I}(f(x_{tgt}) \geq \theta_\tau), \tag{1}$$

where θ_τ is the decision threshold of task τ and $\mathbb{I}(\cdot)$ is the indicator function. The key distinction lies in their training scope. OFO-AD learns f from a task-specific dataset $\mathcal{D}_{train} \subset \mathcal{D}_\tau$, specializing in individual tasks. In contrast, for OFM-AD, the function is trained on a composite dataset that comprises multiple tasks within a single modality, i.e., $\mathcal{D}_{train} \subset \bigcup_{\tau \in T_\mathcal{M}} D_\tau$, where $T_\mathcal{M}$ is the set of tasks under modality $\mathcal{M}$.

In-Context AD

Our proposed ICAD paradigm redefines anomaly detection by leveraging contextual comparison. Let $\mathbb{T}_{train} \subset \mathbb{T}$ represent the subset of tasks observed during training, which can encompass a mixture of modalities. For any task $\tau^* \in \mathbb{T}$ (including unseen tasks where $\tau^* \notin \mathbb{T}_{train}$), we define a reference set $\mathcal{R} = \{r_1, r_2, \cdots, r_K\} \subset \{x \in \mathcal{X}_{\tau^*} : y = 0\}$, consisting of K normal samples that characterize the expected behavior for that task. Given a target sample $x_{tgt} \in \mathcal{X}_{\tau^*}$ and its reference set $\mathcal{R}$, ICAD determines anomalies by computing their contextual discrepancy, and the anomaly detection process can be described as:

$$\mathcal{A}(\mathcal{R}, x_{tgt}; \delta, \theta'_\tau) = \mathbb{I}(\delta(\mathcal{R}, x_{tgt}) \geq \theta'_\tau), \tag{2}$$

where $\delta(\mathcal{R}, x_{tgt})$ measures the dissimilarity between the target and reference samples, and θ'_τ is the task-specific discrepancy threshold. By defining anomalies through dynamic comparison to the provided reference sets, this paradigm decouples the model from a static definition of normality, enabling flexible AD across diverse tasks and modalities.

Methodology

In this section, we propose ICAD-LLM, a unified AD framework that harnesses the powerful in-context learning abilities of Large Language Models to detect anomalies across multiple data modalities. Figure 3 illustrates the overall pipeline of ICAD-LLM, which consists of three key components. First, a *Modality-Aware Encoder* addresses feature alignment by projecting heterogeneous inputs into a unified embedding space. Second, the *Prompt-Guided Representation Module* uses an LLM to extract modality-agnostic representations that are highly sensitive to subtle dissimilarities. Third, the model is trained with a *Contextual Contrastive Learning (CCL)* objective, which sharpens its discriminative power by maximizing the discrepancy for anomalous samples while minimizing it for normal ones. The resulting discrepancy score is then used for final anomaly detection.

Sample Preparation

Before detailing the model architecture, we first describe how raw, heterogeneous data is transformed into a standard-

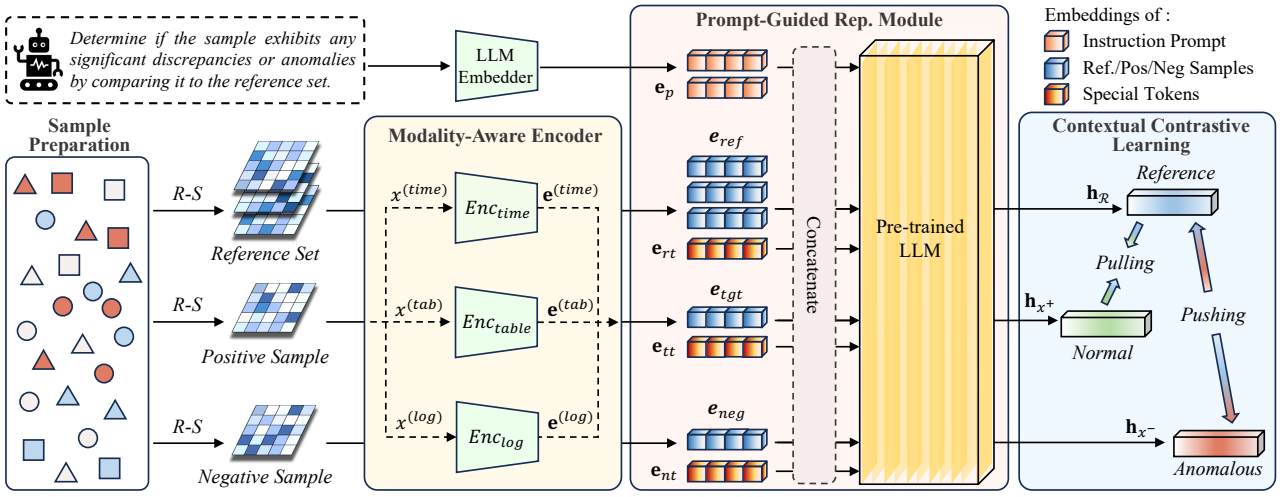

Figure 3: The overall pipeline of ICAD-LLM. A reference set and target samples are first encoded by the Modality-Aware Encoder. The resulting embeddings are then concatenated with prompt and special tokens, and fed into the Prompt-Guided Representation Module to extract context-sensitive representations. Finally, the Contextual Contrastive Learning objective is employed to optimize the model's ability to distinguish between these samples. *R-S* means random selection.

ized "sample" format, which is the fundamental unit that our model can process.

Time Series Processing Let $X_{\text{raw}}^{(\text{time})} \in \mathbb{R}^{L \times d_{\text{raw}}}$ denote a raw time series, where L is the sequence length and d_{raw} is the feature dimension. We define a patching function that segments $X_{\text{raw}}^{(\text{time})}$ into a set of n patches:

$$\left\{x_i^{(\text{time})}\right\}_{i=1}^{n} = \text{Patch}(X_{\text{raw}}^{(\text{time})}). \tag{3}$$

Each patch $x_i^{(\text{time})} \in \mathbb{R}^{p \times d_{\text{raw}}}$ is treated as an individual sample, where p is the patch length.

Tabular Row Processing Let $X_{\text{raw}}^{(\text{tab})} = \{x_{\text{raw},i}^{(\text{tab})}\}_{i=1}^{N}$ be a tabular dataset where each row $x_{\text{raw},i}^{(\text{tab})} \in \mathbb{R}^{F_i}$ has F_i features. To ensure uniform dimensionality, we define a padding-truncation function $\Psi(\cdot)$ that maps each row to a fixed dimension F' with zero-padding or truncation:

$$x_i^{(\text{tab})} = \Psi(x_{\text{raw},i}^{(\text{tab})}) = \begin{cases} [x_{\text{raw},i}^{(\text{tab})}, \mathbf{0}_{F'-F_i}] & \text{if } F_i < F' \\ x_{\text{raw},i}^{(\text{tab})}[0:F'] & \text{if } F_i \geq F' \end{cases} \tag{4}$$

where $\mathbf{0}_{F'-F_i} \in \mathbb{R}^{F'-F_i}$ is a zero vector of length $F' - F_i$, and $x[0:F']$ denotes selecting the first F' elements. Each $x_{\text{raw},i}^{(\text{tab})}$ constitutes a single sample.

Log Sequence Processing Raw log messages $X_{\text{raw}}^{(\text{log})} = \{m_t\}_{t=1}^{T}$ undergo two-stage processing. First, a log parser[1] extracts templates: $S = \{k_t\}_{t=1}^{T} = \text{Parse}(X_{\text{raw}}^{(\text{log})})$, where $k_t \in \mathcal{K}$ represents the log template at time t. Second, temporal windowing partitions S into fixed-size segments: $x_i^{(\text{log})} = S[(i-1)w + 1 : iw]$, where each window of w consecutive log keys forms a sample.

[1]We use Drain3 to parse the logs.

Modality-Aware Encoder

To effectively process heterogeneous samples from various modalities, we employ the Modality-Aware Encoder, which transforms prepared samples from different modalities into a unified embedding space. Given a sample $x^{(\mathcal{M})}$ from modality $\mathcal{M}$, the encoder applies a transformation as:

$$\mathbf{e}^{(\mathcal{M})} = E_{\mathcal{M}}(x^{(\mathcal{M})}), \tag{5}$$

where $\mathbf{e}^{(\mathcal{M})} \in \mathbb{R}^{N^{(\mathcal{M})} \times d_{\text{model}}}$ is the encoded embedding, $N^{(\mathcal{M})}$ is the sequence length, and d_{model} is the embedding dimension. The specific implementations of $E_{\mathcal{M}}$ are as follows: For a time series sample $x^{(\text{time})}$, we first apply instance normalization and then feed the result into a Convolutional Neural Network (CNN) to align the feature dimension:

$$E_{\text{time}}\left(x^{(\text{time})}\right) = \text{CNN}\left(\text{IN}\left(x^{(\text{time})}\right)\right). \tag{6}$$

For a tabular sample $x^{(\text{tab})}$, following the encoding approach of MCM (Yin et al. 2024), the encoder E_{tab} utilizes a two-layer Multilayer Perceptron (MLP) to produce its embedding:

$$E_{\text{tab}}\left(x^{(\text{tab})}\right) = \text{MLP}^{(\text{tab})}\left(x^{(\text{tab})}\right). \tag{7}$$

For a log sample $x^{(\text{log})}$, the encoder E_{log} first uses the LLM's native tokenizer and embedder $\text{Emb}(\cdot)$ to get initial embeddings, which are then refined by a Transformer encoder:

$$E_{\text{log}}\left(x^{(\text{log})}\right) = \text{TransEnc}\left(\text{Emb}\left(x^{(\text{log})}\right)\right). \tag{8}$$

Prompt-Guided Representation Module

At the core of our ICAD model is the Prompt-Guided Representation Module. It leverages a pre-trained Large Language Model (LLM) as the backbone, harnessing its powerful capabilities for contextual reasoning and semantic understanding to produce modality-agnostic representations that

capture the subtle differences between a target sample and its reference set. This module incorporates two key mechanisms: *(1) Instruction-based Priming.* Inspired by prior works (Jin et al. 2024; Yu et al. 2025), we prepend an instruction prompt to the input sequence which explicitly primes the LLM, directing its powerful reasoning abilities towards the specific goal of assessing contextual dissimilarity, rather than general language understanding. *(2) Token-anchored Representation Pooling.* To obtain distinct and high-level representations for both the context and the target, we introduce two special, learnable tokens: [REF_TOK] and [TGT_TOK]. These tokens are strategically inserted into the input sequence to act as designated "pooling anchors," compelling the LLM to aggregate and summarize the information of the reference set and the target sample into their respective token positions.

Formally, let $\mathbf{e}_p \in \mathbb{R}^{L \times d_{\text{model}}}$, where L is the sequence length of prompt tokens, be the embedding of the instruction prompt, $\mathbf{e}_{rt} \in \mathbb{R}^{1 \times d_{\text{model}}}$ and $\mathbf{e}_{tt} \in \mathbb{R}^{1 \times d_{\text{model}}}$ be the corresponding embeddings for the aforementioned [REF_TOK] and [TGT_TOK]. For a given reference set $\mathcal{R}$ and a target sample x, we denote their embeddings produced by the Modality-Aware Encoder as $\mathbf{e}_{ref} \in \mathbb{R}^{N \times d_{\text{model}}}$ and $\mathbf{e}_{tgt} \in \mathbb{R}^{N \times d_{\text{model}}}$, respectively. The final input sequence S is formulated as:

$$S = \text{Concat}\left(\mathbf{e}_p, \mathbf{e}_{ref}, \mathbf{e}_{rt}, \mathbf{e}_{tgt}, \mathbf{e}_{tt}\right), \tag{9}$$

This sequence is then fed into our LLM backbone, and we extract the final-layer hidden states corresponding to the positions of our special tokens. This yields a holistic representation for the reference set, $\mathbf{h}_{\mathcal{R}} \in \mathbb{R}^{d_{\text{model}}}$ and a representation for the target sample, $\mathbf{h}_x \in \mathbb{R}^{d_{\text{model}}}$. These representations encapsulate the essential characteristics of their inputs while being sensitive to their contextual differences, providing a modality-agnostic basis for anomaly detection.

Contextual Contrastive Learning

We propose the Contextual Contrastive Learning (CCL) objective, which creates a clear margin for discrimination as required by the ICAD paradigm by pulling normal samples closer to the representation of their reference set, while pushing anomalous samples further away. To implement this, we formulate the training process around sample triplets, each designed to teach the model a specific aspect of contextual comparison. For each training step, given a source dataset $\mathcal{D}$ from modality $\mathcal{M}$, we construct a triplet $(\mathcal{R}, x^+, x^-)$ as follows:

- Reference Set ($\mathcal{R}$): A set of K normal samples randomly selected from $\mathcal{D}$, defining the normal context.
- Positive Sample (x^+): Another normal sample drawn from $\mathcal{D} \setminus \mathcal{R}$, representing an instance of in-context normality that should be identified as similar to $\mathcal{R}$.
- Simple Negative Sample (x_s^-): A normal sample drawn from a different dataset $\mathcal{D}'$ of the same modality $\mathcal{M}$. This is designed to instill a coarse-grained, foundational discriminative ability to the model.
- Hard Negative Sample (x_h^-): An anomalous sample from the source dataset $\mathcal{D}$, teaching the model to identify subtle, fine-grained deviations that define a true anomaly.

To process a triplet $(\mathcal{R}, x^+, x^-)$ efficiently within a single forward pass, where x^- can be either a simple or a hard negative, we adapt the input sequence from Equation (9) by introducing an additional special token, [NEG_TOK]. Let $\mathbf{e}_{neg} \in \mathbb{R}^{N \times d_{\text{model}}}$ be the embedding of the negative sample, and $\mathbf{e}_{nt} \in \mathbb{R}^{1 \times d_{\text{model}}}$ be the embedding for [NEG_TOK]. The full training sequence is constructed as:

$$\begin{aligned} S_{train} = \text{Concat}\big(&\mathbf{e}_p, \mathbf{e}_{ref}, \mathbf{e}_{rt}, \\ &\mathbf{e}_{tgt}, \mathbf{e}_{tt}, \mathbf{e}_{neg}, \mathbf{e}_{nt}\big) \end{aligned} \tag{10}$$

Notably, we use [TGT_TOK] as the token for the positive sample to maintain consistency with the inference phase.

After this sequence is processed by our model, we extract the final representations for the reference set ($\mathbf{h}_{\mathcal{R}}$), the positive sample ($\mathbf{h}_{x^+}$), and the negative sample ($\mathbf{h}_{x^-}$) from their respective special token positions. Finally, these representations are used to compute the loss function $\mathcal{L}$ of our CCL. Let $s(\cdot, \cdot)$ denote the cosine similarity, then $\mathcal{L}$ is defined as:

$$\mathcal{L} = \max\left(s(\mathbf{h}_{\mathcal{R}}, \mathbf{h}_{x^+}) - s(\mathbf{h}_{\mathcal{R}}, \mathbf{h}_{x^-}) + \alpha, 0\right), \tag{11}$$

where $\alpha > 0$ is a margin hyperparameter that enforces a minimum distance between positive and negative pairs. By minimizing this loss, the model is explicitly trained to produce a low discrepancy score for contextually similar pairs and a high score for dissimilar ones, directly fulfilling the discriminative learning requirement of the ICAD paradigm.

Anomaly Detection During Inference

During model inference, the setup is simplified. Given a reference set $\mathcal{R}$ and a test sample x_{test}, we use the ICAD-LLM model to compute their representations $\mathbf{h}_{\mathcal{R}}$ and $h_{\mathbf{x}_{test}}$. The discrepancy core $\delta\left(\mathcal{R}, x_{test}\right)$ is then calculated as the distance between them:

$$\delta\left(\mathcal{R}, x_{test}\right) = \frac{1 - s(\mathbf{h}_{\mathcal{R}}, \mathbf{h}_{x_{test}})}{2}, \tag{12}$$

This score is compared against a pre-defined threshold. If the score exceeds this threshold, the sample x_{test} is classified as an anomaly relative to the context provided by $\mathcal{R}$.

Experiment

Experimental Setup

Datasets Our study employs a diverse collection of AD datasets spanning multiple modalities. For time series AD, we select five prominent datasets: SMD (Su et al. 2019b), PSM (Abdulaal, Liu, and Lancewicki 2021), SWaT (Mathur and Tippenhauer 2016), MSL, and SMAP (Hundman et al. 2018). To evaluate performance on tabular data, we incorporate 18 real-world datasets sourced from ADBench (Han et al. 2022). Furthermore, for log AD, we select four widely used datasets, including BGL, Thunderbird, Liberty2, and Spirit2 (Oliner and Stearley 2007).

Metrics Our evaluation strategy employs distinct metrics tailored to the characteristics of different data modalities. For both tabular and log datasets, we utilize AUROC as our evaluation metric. For time series, we follow prior works (Shen, Li, and Kwok 2020; Xu et al. 2022) and employ F1-score with point adjustment.

Modality / Dateset	Metric	Task-Specific AD Methods					Universal AD Methods			
Time Series		Anoamly.	DLinear	TimesNet	OneFitsAll	Ours	NeuTraL.	UniAD	ACR	Ours
SMD	F1	85.68	79.34	85.94	86.92	**88.47**	81.47	84.32	74.38	**88.24**
MSL		84.12	85.41	85.78	82.48	**86.52**	79.68	81.99	76.43	**85.15**
SMAP		71.57	70.39	72.07	72.84	**75.27**	64.29	**74.02**	69.48	71.95
SWAT		84.29	89.25	92.37	94.27	**94.55**	77.43	79.38	**90.7**	87.98
PSM		82.36	93.7	97.33	97.16	**97.64**	91.64	92.84	89.53	**96.97**
Average		81.6	83.62	86.7	86.73	**88.49**	78.9	82.51	80.1	**85.66**
Tabular		IForest	DAGMM	GOAD	MCM	Ours	NeuTraL.	UniAD	ACR	Ours
Cardio	AUROC	79.67	66.61	86.27	94.34	**94.69**	63.75	62.71	65.51	**91.03**
Campaign		69.77	75.6	83.28	87.32	**87.44**	56.94	50.8	53.05	**85.31**
Fraud		73.63	81.47	78.09	**93.23**	92.64	67.07	74.38	65.75	**85.26**
HTTP		86.7	92.47	96.78	97.77	**98.14**	**94.18**	88.61	86.75	91.71
Optdigits		79.73	64.29	79.62	97.32	**97.88**	63.26	63.32	61.14	**90.23**
Shuttle		93.18	90.11	96.07	**99.31**	98.74	88.14	90.75	92.24	**95.34**
SMTP		86.73	88.94	90.08	91.47	**92.46**	**90.41**	83.04	83.45	86.52
Wbc		84.51	77.86	85.31	97.89	**99.06**	85.17	76.91	77.06	**97.94**
Average		81.74	79.67	86.94	94.83	**95.13**	76.12	73.82	73.12	**90.42**
Log		LogCluster	DeepLog	LogAnomaly	LogBert	Ours	NeuTraL.	UniAD	ACR	Ours
BGL	AUROC	83.72	90.29	82.35	93.66	**95.32**	75.39	77.24	84.66	**92.79**
Thunderbird		74.28	91.88	93.24	92.37	**94.84**	67.36	82.31	78.75	**85.2**
Liberty2		83.24	86.27	93.63	94.29	**98.47**	72.55	86.29	84.06	**88.69**
Spirit2		88.79	95.25	92.89	95.27	**97.24**	81.49	79.17	87.52	**90.44**
Average		82.51	90.92	90.53	93.9	**96.47**	74.2	81.25	83.75	**89.28**

Table 1: Performance comparison of different anomaly detection methods on time series, tabular, and log data. The best and second-best results in each category are shown in bold and with an underline, respectively.

Implementation Details We use Qwen2.5-0.5B (Team 2024) as the pre-trained backbone. During training, $K = 5$ samples are randomly selected to form the reference set ($\mathcal{R}$), with an 8:2 ratio of simple to hard negatives. The model is trained for 5 epochs with a learning rate of 1e-5, sampling 200k instances per epoch across modalities.

Baseline Methods To provide a comprehensive evaluation of ICAD-LLM's performance, we compare it against two distinct categories of methods: task-specific baselines and universal baselines. Task-specific models are exclusively trained on their respective single-modality datasets to optimize performance within their specialized domain. For time series AD, we include Anomaly Transformer (Xu et al. 2022), DLinear (Zeng et al. 2023), TimesNet (Wu et al. 2023), and OneFitsAll (Zhou et al. 2023); for tabular AD, we evaluate against Isolation Forest (Liu, Ting, and Zhou 2008), DAGMM (Zong et al. 2018), GOAD (Bergman and Hoshen 2020), and MCM (Yin et al. 2024); and for log AD, we compare with LogCluster (Lin et al. 2016), DeepLog (Du et al. 2017), LogAnomaly (Meng et al. 2019), and LogBert (Guo, Yuan, and Wu 2021). To assess the capability of handling diverse data types and tasks within a single framework, we also compare ICAD-LLM against universal AD methods including NeuTraL AD (Qiu et al. 2021), UniAD (You et al. 2022) and ACR (Li et al. 2023). They are trained across all datasets used in our experiment, mirroring the training scope of ICAD-LLM to ensure a fair comparison.

Main Results

Table 1 summarizes the comprehensive AD performance of ICAD-LLM against competitive baselines across multiple modalities, with results structured to highlight two key comparisons: task-specific and universal AD. Notably, Table 1 summarizes the results on 8 key tabular datasets, while the full results of all 18 datasets are provided in the appendix.

Comparison with Task-Specific Methods We first evaluate ICAD-LLM against task-specific baselines within each modality. While each baseline model is trained individually for a single dataset, ICAD-LLM uses a single model, trained only once on a composite dataset comprising all tasks within that modality. As shown in Table 1, ICAD-LLM consistently achieves competitive or superior performance across nearly all benchmarks (e.g., $\uparrow$ 4.18 on Liberty2 compared to the second-best result). Our approach proves that even a single model can develop a sufficient understanding of anomalies to specialized counterparts.

Comparison with Universal Methods We evaluate the performance of universal anomaly detection methods trained jointly on datasets from all three modalities—time series, tabular, and log data—using a single shared model. Under this challenging setting, ICAD-LLM delivers performance that closely approaches and occasionally even surpasses strong task-specific baselines. In contrast, existing universal AD methods yield results that are generally inferior to task-specific approaches and often lack reliabil-

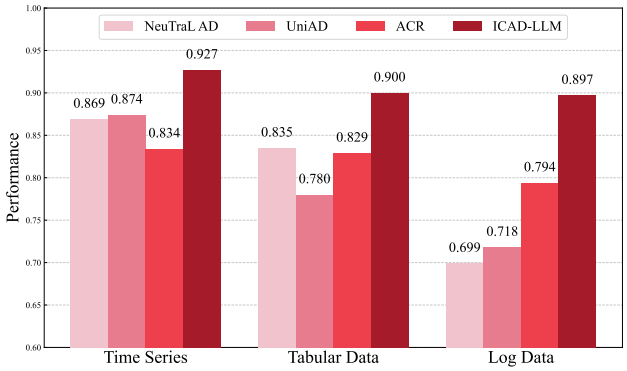

Figure 4: Generalization performance comparison different methods on unseen datasets across three modalities.

ity across diverse datasets. These observations demonstrate ICAD-LLM's strong adaptability and its practical value in meeting the growing demand for efficient, scalable anomaly detection solutions capable of handling diverse modalities within a single model.

Generalization Experiment

To evaluate ICAD-LLM's ability to generalize to unseen data, we conduct experiments where the model is tested on datasets that were entirely excluded from training. This evaluation comprises one time series dataset, four tabular datasets, and one log dataset. For the tabular modality, we report the average performance. Experiment details and complete results are provided in the appendix. As shown in Figure 4, ICAD-LLM maintains strong performance on these unseen datasets, outperforming all baselines across every data modality. In contrast, the baselines exhibit inconsistent performance across different data types, highlighting their limited adaptability. These results validate the robustness and effective generalization capability of our model.

Sensitivity Analysis

To thoroughly understand the contribution of key design choices in ICAD-LLM, we conduct sensitivity analysis to investigate two critical factors: the number of samples in the reference set and the total volume of training data. Detailed experimental setups and the complete results of this analysis are provided in the appendix.

Impact of Reference Set Size We assess the influence of the reference set size, K, by varying it across a range of values, with a focus on smaller sizes ($K = 1, 2, 3, 5$) and also including larger values ($K = 7, 10$) to observe the trend. As shown in Figure 5(a), it is observed that as K increases, the average performance initially rises rapidly. However, beyond $K = 5$, the performance improvement becomes notably slower. This phenomenon may be attributed to the representativeness of the reference set: smaller reference set sizes may not adequately capture the common characteristics of normal instances, while larger sizes yield diminishing marginal returns as the informative content becomes saturated. Consequently, $K = 5$ is selected as the reference set size for all other experiments in our study.

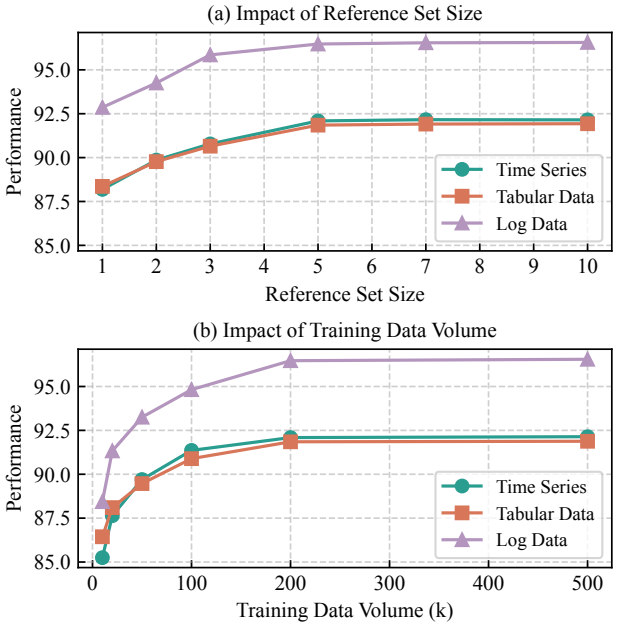

Figure 5: Impact of (a) reference set size and (b) total training data volume on the performance of ICAD-LLM.

Impact of Total Training Data volume We evaluate six different scales of total training data volumes, ranging from 10k to 500k samples, sourced from all datasets across the various modalities. As presented in Figure 5(b), larger data volumes consistently lead to improved model performance. However, beyond 200k samples, the performance gains become marginal while the training cost increases significantly. Therefore, considering this clear performance-cost trade-off, we adopt 200k samples as the standard training volume in our work.

Conclusion

In this paper, we introduced a new AD paradigm, In-Context Anomaly Detection (ICAD), which reframes AD from memorizing a static normal distribution to performing dynamic in-context comparison. Our proposed model, ICAD-LLM, realizes this paradigm by leveraging a Large Language Model to learn a general discrepancy function, enabling it to be trained once and then applied across diverse modalities and unseen tasks without task-specific retraining. Extensive experiments validate this approach, demonstrating that ICAD-LLM achieves competitive performance against specialized methods while exhibiting robust generalization. This work advances the vision of a One-for-ALL AD framework, offering a viable path for developing more scalable and adaptable systems for real-world applications.

Acknowledgments

Jingyuan Wang's work was partially supported by the National Natural Science Foundation of China (No. 72171013, 72222022, 72242101) and the Fundamental Research Funds for the Central Universities (JKF-2025017226182). Juhua Pu's work was partially supported by the National Natural Science Foundation of China (No. 62577006).

References

Abdulaal, A.; Liu, Z.; and Lancewicki, T. 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2485–2494.

Bergman, L.; and Hoshen, Y. 2020. Classification-Based Anomaly Detection for General Data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93–104.

Du, M.; Li, F.; Zheng, G.; and Srikumar, V. 2017. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 1285–1298.

Grubbs, F. E. 1969. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1): 1–21.

Guo, H.; Yuan, S.; and Wu, X. 2021. Logbert: Log anomaly detection via bert. In *2021 international joint conference on neural networks (IJCNN)*, 1–8. IEEE.

Han, S.; Hu, X.; Huang, H.; Jiang, M.; and Zhao, Y. 2022. Adbench: Anomaly detection benchmark. *Advances in neural information processing systems*, 35: 32142–32159.

He, H.; Bai, Y.; Zhang, J.; He, Q.; Chen, H.; Gan, Z.; Wang, C.; Li, X.; Tian, G.; and Xie, L. 2024. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2404.06564*.

Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; and Soderstrom, T. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 387–395.

Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19606–19616.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.; Liang, Y.; Li, Y.; Pan, S.; and Wen, Q. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Li, A.; Qiu, C.; Kloft, M.; Smyth, P.; Rudolph, M.; and Mandt, S. 2023. Zero-shot anomaly detection via batch normalization. *Advances in Neural Information Processing Systems*, 36: 40963–40993.

Li, K.-L.; Huang, H.-K.; Tian, S.-F.; and Xu, W. 2003. Improving one-class SVM for anomaly detection. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, volume 5, 3077–3081 Vol.5.

Lin, Q.; Zhang, H.; Lou, J.; Zhang, Y.; and Chen, X. 2016. Log clustering based problem identification for online service systems. In Dillon, L. K.; Visser, W.; and Williams, L. A., eds., *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016 - Companion Volume*, 102–111. ACM.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*, 413–422.

Liu, S.; Yao, D.; Fang, L.; Li, Z.; Li, W.; Feng, K.; Ji, X.; and Bi, J. 2024. Anomalyllm: Few-shot anomaly edge detection for dynamic graphs using large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, 785–790. IEEE.

Liu, Z.; Zhou, Y.; Xu, Y.; and Wang, Z. 2023. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20402–20411.

Mathur, A. P.; and Tippenhauer, N. O. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, 31–36. IEEE.

Meng, W.; Liu, Y.; Zhu, Y.; Zhang, S.; Pei, D.; Liu, Y.; Chen, Y.; Zhang, R.; Tao, S.; Sun, P.; et al. 2019. Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In *IJCAI*, volume 19, 4739–4745.

Oliner, A.; and Stearley, J. 2007. What supercomputers say: A study of five system logs. In *37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07)*, 575–584. IEEE.

Qiu, C.; Pfrommer, T.; Kloft, M.; Mandt, S.; and Rudolph, M. 2021. Neural transformation learning for deep anomaly detection beyond images. In *International conference on machine learning*, 8703–8714. PMLR.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. V. 2022. Towards Total Recall in Industrial Anomaly Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 14298–14308. IEEE.

Shen, L.; Li, Z.; and Kwok, J. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in neural information processing systems*, 33: 13016–13026.

Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; and Pei, D. 2019a. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In Teredesai, A.; Kumar, V.; Li, Y.; Rosales, R.; Terzi, E.; and

Karypis, G., eds., *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2828–2837. ACM.

Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; and Pei, D. 2019b. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2828–2837.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xu, H.; Chen, W.; Zhao, N.; Li, Z.; Bu, J.; Li, Z.; Liu, Y.; Zhao, Y.; Pei, D.; Feng, Y.; et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, 187–196.

Xu, J.; Wu, H.; Wang, J.; and Long, M. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yang, Y.; Zhang, C.; Zhou, T.; Wen, Q.; and Sun, L. 2023. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3033–3045.

Yao, X.; Chen, Z.; Gao, C.; Zhai, G.; and Zhang, C. 2024. Resad: A simple framework for class generalizable anomaly detection. *Advances in Neural Information Processing Systems*, 37: 125287–125311.

Yao, X.; Zhang, C.; Li, R.; Sun, J.; and Liu, Z. 2023. One-for-All: Proposal Masked Cross-Class Anomaly Detection. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 4792–4800. AAAI Press.

Yin, J.; Qiao, Y.; Zhou, Z.; Wang, X.; and Yang, J. 2024. Mcm: Masked cell modeling for anomaly detection in tabular data. In *The Twelfth International Conference on Learning Representations*.

You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; and Le, X. 2022. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35: 4571–4584.

Yu, X.; Wang, J.; Yang, Y.; Huang, Q.; and Qu, K. 2025. BIGCity: A Universal Spatiotemporal Model for Unified Trajectory and Traffic State Data Analysis. In *41st IEEE International Conference on Data Engineering, ICDE 2025, Hong Kong, May 19-23, 2025*, 4455–4469. IEEE.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.

Zong, B.; Song, Q.; Min, M. R.; Cheng, W.; Lumezanu, C.; Cho, D.; and Chen, H. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.

Appendix

Implementation Details

Model Architecture and Hyperparameters All experiments were conducted on a server equipped with 8 NVIDIA RTX 4090 GPUs. We use Qwen2.5-0.5B as the base model. The model was trained using the Adam optimizer with a learning rate of 1e-5, and a batch size of 64, 256, and 32 for time series, tabular data, and log data, respectively. We trained the model for a total of 5 epochs on the 200k combined dataset. Our implementation is based on Python 3.10, PyTorch 2.6.0, and the Transformers 4.47.1.

Prompt Formulation To guide the LLM's contextual reasoning process, we designed a specific instruction prompt for the Prompt-Guided Representation Module. This prompt structures the task for the model, instructing it to perform a comparison rather than a simple language understanding. The prompt template used in our experiments is as follows:

> *"Determine if the sample exhibits any significant discrepancies or anomalies by comparing it to the reference set."*

This instruction serves as a prefix to the concatenated embeddings of the target and reference samples, forming the final input for the LLM.

Data Splitting Details

- Task-specific Setting: For a dataset i within a single modality, the sampling probability p_i is proportional to its size N_i (number of samples): $p_i = N_i / \sum_j N_j$. For time-series datasets, to avoid severe under-sampling of smaller sets, we adjust the effective size to $N_i' = max(N_i, 250k \text{ time points})$, preventing smaller datasets from being under-sampled.
- Universal Setting: To ensure balanced training across modalities, we first select a modality (Tabular, Time-series, or Log) with uniform probability (1/3). Then, within the chosen modality, we sample triplets from its constituent datasets using the same proportional-to-size strategy as in the task-specific setting.

Complete Results of Main Experiment

For brevity, the main paper presents results on a representative subset of 8 tabular datasets. Table 2 provides the complete performance comparison of ICAD-LLM against baselines on all 18 tabular datasets. As shown, ICAD-LLM consistently ranks as the best or second-best method across all datasets, supporting the conclusions drawn in the main text.

Generalization Experiment Details

Experimental Setups In the generalization experiment, we test the models on datasets entirely excluded from the training pool to evaluate their zero-shot adaptation capability. The selected datasets (PSM for time series; HTTP, Shuttle, SMTP, and Wbc for tabular data; BGL for log data) are widely used benchmarks representing diverse real-world scenarios. The training set composition was adjusted to maintain the total volume at 200k samples, with the reference set size K fixed at 5.

Detailed Results Table 3 presents the detailed scores for each method on the held-out datasets. The results show that ICAD-LLM maintains high performance across all modalities, whereas the performance of baseline methods varies significantly.

Sensitivity Analysis Details

Impact of Reference Set Size This experiment investigates the model's sensitivity to the reference set size. We vary K within the range of $\{1, 2, 3, 5, 7, 10\}$ while keeping the total training data volume fixed at 200k. Table 4 provides the detailed results for each dataset.

Impact of Training Data Volume This experiment analyzes the effect of the total training data volume on model performance. We vary the volume from 10k to 500k samples, with the reference set size K fixed at 5. Table 5 shows the detailed results. The trend of diminishing returns observed in the main paper is evident across most individual datasets, confirming that 200k samples is a data-efficient choice.

Modality / Dateset	Metric	Task-Specific AD Methods					Universal AD Methods			
Time Series		Anoamly.*	DLinear	TimesNet	OneFitsAll	Ours	NeuTraL.	UniAD	ACR	Ours
SMD	F1	85.68	79.34	85.94	86.92	**88.47**	81.47	84.32	74.38	**88.24**
MSL		84.12	85.41	85.78	82.48	**86.52**	79.68	81.99	76.43	**85.15**
SMAP		71.57	70.39	72.07	72.84	**75.27**	64.29	**74.02**	69.48	71.95
SWAT		84.29	89.25	92.37	94.27	**94.55**	77.43	79.38	**90.70**	87.98
PSM		82.36	93.70	97.33	97.16	**97.64**	91.64	92.84	89.53	**96.97**
Average		81.60	83.62	86.70	86.73	**88.49**	78.90	82.51	80.10	**85.66**
Tabular		IForest	DAGMM	GOAD	MCM	Ours	NeuTraL.	UniAD	ACR	Ours
Breastw	AUROC	86.20	74.07	83.28	99.45	**99.67**	67.64	80.72	72.96	**91.73**
Cardio		79.67	66.61	86.27	94.34	**94.69**	63.75	62.71	65.51	**91.03**
Campaign		69.77	75.60	83.28	87.32	**87.44**	56.94	50.80	53.05	**85.31**
Cardiotocography		70.53	79.18	73.92	78.84	**81.21**	52.32	67.55	61.39	**74.86**
Fraud		73.63	81.47	78.09	**93.23**	92.64	67.07	74.38	65.75	**85.26**
Glass		64.80	68.57	65.09	69.34	**70.50**	58.01	63.96	52.35	**66.95**
HTTP		86.70	92.47	96.78	97.77	**98.14**	**94.18**	88.61	86.75	91.71
Ionosphere		75.59	69.49	89.26	96.91	**99.18**	75.75	66.08	80.94	**97.64**
Mammography		81.69	73.40	81.77	89.91	**91.93**	76.63	69.76	75.33	**84.58**
Optdigits		79.73	64.29	79.62	97.32	**97.88**	63.26	63.32	61.14	**90.23**
Pima		65.36	59.65	71.80	74.86	**75.38**	61.16	58.05	59.38	**68.39**
Pendigits		84.97	68.07	86.50	97.33	**99.68**	54.29	58.89	73.83	**96.90**
Satellite		80.48	72.84	73.91	78.57	**80.93**	67.98	**80.76**	74.95	75.13
Satimage-2		91.45	87.54	97.41	97.32	**97.72**	78.86	82.49	89.92	**95.35**
Shuttle		93.18	90.11	96.07	**99.31**	98.74	88.14	90.75	92.24	**95.34**
SMTP		86.73	88.94	90.08	91.47	**92.46**	**90.41**	83.04	83.45	86.52
Wbc		84.51	77.86	85.31	97.89	**99.06**	85.17	76.91	77.06	**97.94**
Wine		63.79	86.48	83.43	93.88	**96.35**	80.54	87.93	88.84	**89.45**
Average		78.82	76.48	83.44	90.84	**91.87**	71.23	72.60	73.05	**86.91**
Log		LogCluster	DeepLog	LogAnomaly	LogBert	Ours	NeuTraL.	UniAD	ACR	Ours
BGL	AUROC	83.72	90.29	82.35	93.66	**95.32**	75.39	77.24	84.66	**92.79**
Thunderbird		74.28	91.88	93.24	92.37	**94.84**	67.36	82.31	78.75	**85.20**
Liberty2		83.24	86.27	93.63	94.29	**98.47**	72.55	86.29	84.06	**88.69**
Spirit2		88.79	95.25	92.89	95.27	**97.24**	81.49	79.17	87.52	**90.44**
Average		82.51	90.92	90.53	93.90	**96.47**	74.20	81.25	83.75	**89.28**

* We replace the joint criterion in Anomaly Transformer with reconstruction error for consistency with other baseline methods.

Table 2: Performance comparison of different anomaly detection methods on time series, tabular, and log data. The best and second-best results in each category are shown in bold and with an underline, respectively.

Modality	Dataset(s)	Metrics	NeuTralAD	UniAD	ACR	ICAD-LLM
Time Series	PSM	F1	86.94	87.44	83.44	**92.65**
Tabular	HTTP	AUROC	**89.89**	76.73	86.35	88.37
	Shuttle		83.75	87.07	91.37	**91.64**
	SMTP		**84.10**	75.31	78.86	81.99
	Wbc		76.21	72.87	74.99	**97.60**
	Average		83.49	78.00	82.90	**89.96**
Log	BGL	AUROC	69.90	71.76	79.39	**89.65**

Table 3: Generalization performance on unseen datasets. The model is tested on datasets entirely excluded from the training pool. The best results are in bold and the second-best are underlined.

Modality	Datasets	Metrics	K=1	K=2	K=3	K=5	K=7	K=10
Time Series	SMD	F1	87.28	86.96	86.99	88.47	**88.54**	88.53
	MSL		82.91	85.94	86.07	86.52	86.65	**86.66**
	SMAP		90.60	90.68	91.18	93.27	**93.31**	93.30
	SWAT		89.62	92.05	93.86	**94.55**	**94.55**	94.52
	PSM		90.53	93.68	95.81	97.64	**97.75**	97.72
	Average		88.19	89.86	90.78	92.09	**92.16**	92.15
Tabular	Breastw	AUROC	96.33	98.30	98.38	99.67	99.65	**99.7**
	Cardio		89.27	90.32	92.00	94.34	94.34	**94.36**
	Campaign		83.65	85.04	86.11	87.44	**87.49**	**87.49**
	Cardiotocography		78.02	79.8	**81.47**	81.21	81.22	81.20
	Fraud		91.85	92.49	92.65	92.64	**92.81**	92.79
	Glass		68.68	68.48	70.36	70.50	70.66	**70.69**
	HTTP		94.68	97.01	98.14	98.14	98.18	**98.23**
	Ionosphere		96.61	97.43	97.48	99.18	99.34	**99.36**
	Mammography		86.53	89.94	90.59	91.93	92.04	**92.08**
	Optdigits		95.83	95.97	96.07	97.88	97.90	**97.96**
	Pima		73.31	74.05	74.31	**75.38**	75.37	75.36
	Pendigits		95.37	96.62	97.13	99.68	**99.72**	99.71
	Satellite		77.91	78.20	78.51	80.93	80.96	**81.00**
	Satimage-2		94.64	94.48	96.45	97.72	97.79	**97.82**
	Shuttle		93.00	95.82	96.86	**98.74**	**98.74**	98.73
	SMTP		86.75	89.52	90.78	92.46	92.54	**92.55**
	Wbc		95.21	97.09	98.62	99.06	99.22	**99.29**
	Wine		92.83	95.25	95.85	96.35	96.49	**96.50**
	Average		88.36	89.77	90.65	91.85	91.91	**91.93**
Log	BGL	AUROC	93.21	93.35	95.27	95.32	95.41	**95.45**
	Thunderbird		91.67	93.62	93.75	94.84	94.91	**94.92**
	Liberty		95.56	96.44	98.67	98.47	**98.76**	98.75
	spirit		91.01	93.58	95.69	**97.24**	97.07	97.13
	Average		92.86	94.25	95.85	96.47	96.54	**96.56**

Table 4: Sensitivity analysis with respect to the reference set size K. Performance is reported for each dataset as K varies. The best and second-best results for each dataset are in bold and underlined, respectively.

Modality	Datasets	Metrics	10K	20K	50K	100K	200K	500K
Time Series	SMD	F1	81.71	85.63	85.31	87.39	88.47	**88.48**
	MSL		78.56	79.34	83.81	85.46	86.52	**86.60**
	SMAP		87.94	92.25	91.95	92.87	**93.27**	93.25
	SWAT		85.24	88.34	92.39	94.27	94.55	**94.72**
	PSM		92.78	92.64	95.04	96.82	**97.64**	**97.64**
	Average		85.25	87.64	89.71	91.36	92.09	**92.14**
Tabular	Breastw	AUROC	96.19	96.59	99.55	99.57	**99.67**	**99.67**
	Cardio		83.35	87.11	90.44	92.84	94.34	**94.54**
	Campaign		77.89	81.91	84.26	85.98	87.44	**87.46**
	Cardiotocography		71.64	75.39	78.66	81.20	81.21	**81.25**
	Fraud		88.34	90.75	90.41	91.66	**92.64**	92.62
	Glass		65.71	67.46	67.44	69.81	**70.50**	70.44
	HTTP		92.71	93.17	93.98	95.95	98.14	**98.20**
	Ionosphere		95.85	95.97	96.85	97.98	**99.18**	99.10
	Mammography		84.85	87.19	89.62	91.06	91.93	**92.09**
	Optdigits		94.06	95.26	97.15	**98.06**	97.88	97.86
	Pima		70.15	72.47	73.00	73.81	75.38	**75.49**
	Pendigits		95.73	96.89	98.81	99.47	99.68	**99.82**
	Satellite		77.85	79.05	78.91	79.69	**80.93**	80.85
	Satimage-2		94.93	94.53	95.70	97.47	97.72	**97.74**
	Shuttle		93.45	93.82	95.51	96.64	**98.74**	98.68
	SMTP		87.88	90.91	91.01	91.09	92.46	**92.64**
	Wbc		93.28	93.24	95.42	97.49	99.06	**99.17**
	Wine		92.06	94.03	93.79	96.32	**96.35**	96.32
	Average		86.44	88.10	89.47	90.89	91.85	**91.88**
Log	BGL	AUROC	84.89	87.06	90.24	92.83	95.32	**95.42**
	Thunderbird		86.78	90.97	94.06	94.52	94.84	**94.96**
	Liberty		91.77	94.55	94.49	96.92	98.47	**98.54**
	spirit		90.30	92.75	94.21	95.01	97.24	**97.27**
	Average		88.44	91.33	93.25	94.82	96.47	**96.55**

Table 5: Sensitivity analysis with respect to the total training data volume. Performance is reported for each dataset across different training data sizes. The best and second-best results for each dataset are in bold and underlined, respectively.