TASKBRIDGE: BRIDGING UNSUPERVISED TABULAR ANOMALY DETECTION AND IN-CONTEXT LEARNING VIA VIRTUAL TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised tabular anomaly detection (TAD) aims to identify anomalous rows in tabular data using normal training samples. While conventional methods rely on dataset-specific training and configuration search, recent tabular foundation models (TFMs) enable zero-shot anomaly detection on unseen datasets via in-context learning. Most TFM-based approaches, however, require anomaly-specific pretraining from scratch, making detection inherently dependent on synthetic TAD-specific priors and costly to update. Some approaches instead repurpose pretrained general-purpose TFMs for TAD to avoid this burden, but rely on computationally expensive formulations with restrictive anomaly inductive biases. In this work, we introduce TASKBRIDGE, a new framework that efficiently repurposes pretrained general-purpose TFMs for unsupervised TAD by constructing virtual supervised tasks that directly recast anomaly detection as supervised in-context inference of TFMs. The resulting virtual tasks induce predictive structures under which normal queries and their target pairs receive high support, whereas anomalies tend to violate the induced structures and receive lower support, providing direct anomaly evidence. Across 790 real-world datasets, TASKBRIDGE consistently outperforms 30 baselines, including state-of-the-art TFM-based approaches, without anomaly-specific TFM pretraining or dataset-specific model optimization.

1 INTRODUCTION

Tabular anomaly detection (TAD) plays a crucial role in various real-world applications, including finance (Al-Hashedi & Magalingam, 2021), healthcare (Fernando et al., 2021), and security (Ahmad et al., 2021). Tabular samples lack explicit structure, unlike other domains such as images and text, making anomalous patterns more challenging to identify. Various methods have been proposed for TAD, ranging from statistical and conventional machine learning approaches to deep learning-based methods (Liu et al., 2008; Breunig et al., 2000; Livernoche et al., 2024).

In TAD, anomalous observations are often rare and difficult to label in advance, making it common to use only normal data for training (Graham et al., 2023; Maziarka et al., 2021). In this *unsupervised* or *one-class* setting, selecting a suitable detector or tuning its hyperparameters (HPs) is highly challenging since no guiding signals are provided; conventional HP searches or dataset-specific tuning cannot be reliably guided without validation data (Ma et al., 2023; Ding et al., 2022).

To address this limitation, recent work has studied tabular foundation models (TFMs) for unsupervised TAD as a way to reduce reliance on dataset-specific training and HP tuning. A prominent line of work is to use *prior-data fitted networks* (PFNs) (Shen et al., 2025; Ding et al., 2026b; Marszałek et al., 2026). After pretraining over synthetic tasks, PFNs perform zero-shot inference through *in-context learning* (ICL) by inferring the task structure from a given context and amortizing Bayesian posterior-predictive inference for unseen queries (Müller et al., 2024).

There are two notable approaches to using TFMs for unsupervised TAD. The first develops TFMs pretrained exclusively for TAD (Shen et al., 2025; Ding et al., 2026b). During pretraining, unlabeled normal samples form the context, while synthetic anomalies are introduced as queries, allowing the model to infer whether a query is anomalous conditioned on the normal context. By amortizing TAD

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

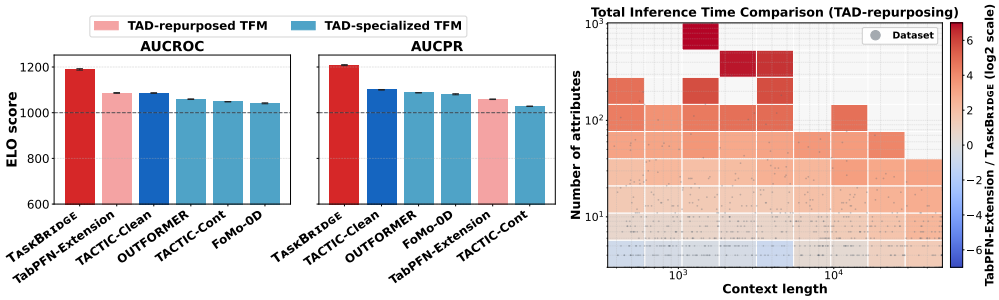


Figure 1: (Left) Elo scores of TFM-based TAD methods on ODDBench; darker colors indicate better performance. TASKBRIDGE achieves the highest Elo scores for both AUCROC and AUCPR. Full baseline comparisons are provided in Appendix C.1. (Right) Inference-time comparison between the TabPFN-Extension and TASKBRIDGE, with red indicating that TASKBRIDGE is faster. TASKBRIDGE scales more efficiently as both context length and dimensionality increase.

across diverse episodic tasks with varying synthetic anomaly mechanisms, these models enable in-context anomaly detection on unseen datasets. We refer to them as *TAD-specialized TFMs*.

Complementing TAD-specialized TFMs, TabPFN (Grinsztajn et al., 2026b), which is a representative TFM pretrained for general-purpose tabular modeling, supports a TAD extension that harnesses its pretrained predictive capabilities for anomaly detection through autoregressive joint-density estimation. Such repurposing avoids TAD-specific pretraining and can directly benefit from advances in general-purpose TFMs. We refer to approaches of this kind as *TAD-repurposed TFMs*.

Both directions have notable limitations. TAD-specialized TFMs, relying on AD-centric pretraining with designed synthetic priors, require redesigning the priors and pretraining strategy followed by model retraining, both to expand anomaly coverage and to incorporate advances in increasingly capable TFM backbones. The existing TAD-repurposed approach avoids such burdens, but its autoregressive formulation requires repeated attribute-wise conditional predictions, resulting in substantial inference overhead as dimensionality increases, while reducing the notion of anomalousness to low estimated joint likelihood. We discuss these limitations in greater detail in Section 2.2.

To address these limitations, we propose TASKBRIDGE, a new TAD framework that efficiently repurposes pretrained general-purpose TFMs by constructing *virtual supervised tasks* from unlabeled, normality-aware training data. These tasks induce a predictive structure through the in-context inference over the virtually labeled context, under which normal query–target pairs receive high predictive support, whereas anomalous pairs tend to violate the induced structure and receive lower support, directly providing evidence of anomalies. To generalize this mechanism across hundreds of datasets, we create task templates that probe different aspects of normal structure and apply data-adaptive task selection to retain tasks that best satisfy the AD-oriented properties for each dataset.

TASKBRIDGE offers several advantages: (i) it enables zero-shot AD inference on unseen datasets, without dataset-specific training or tuning, while requiring neither TAD-specialized pretraining nor a dedicated backbone; (ii) as shown in the left panel of Figure 1, it achieves stronger overall performance than existing TAD baselines, including diverse TFM-based approaches, across 790 real-world datasets; and (iii) as shown in the right panel of Figure 1, TASKBRIDGE remains substantially more efficient than the TabPFN-Extension as both dataset size and dimensionality increase, demonstrating a more scalable and practical way to repurpose general-purpose TFMs for unsupervised TAD.

2 PRELIMINARIES AND RELATED WORKS

2.1 GENERAL-PURPOSE TABULAR FOUNDATION MODELS

Pretraining. Let $\mathbf{a} = (\mathrm{a}_1, \ldots, \mathrm{a}_d)$ denote the vector of attribute random variables, and let y denote the target random variable. A realized sample is written as $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^\top \in \mathbb{X} \subseteq \mathbb{R}^d$. Let $\tau \in \mathcal{T}$ denote a *task* and let $p(\tau)$ denote the *prior* used to sample pretraining tasks. For each pretraining episode, a task τ is first sampled from $p(\tau)$, after which a labeled context and a query sample are drawn from the corresponding task-specific distribution. Under the task prior, the

Bayesian posterior predictive distribution (PPD) is defined as

$$\tau \sim p(\tau), \qquad \mathcal{D}_{\text{ctx}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}, \qquad \mathcal{D}_{\text{ctx}} \cup \{(\boldsymbol{x}_q, y_q)\} \overset{\text{i.i.d.}}{\sim} P_\tau,$$
$$p(y_q \mid \boldsymbol{x}_q, \mathcal{D}_{\text{ctx}}) = \int_{\mathcal{T}} p_\tau(y_q \mid \boldsymbol{x}_q)\, p(\tau \mid \mathcal{D}_{\text{ctx}})\; d\tau. \tag{1}$$

A general-purpose TFM q_θ, typically instantiated as a transformer (Vaswani et al., 2017), is trained to approximate the PPD by minimizing the prior-data negative log-likelihood:

$$\mathcal{L}_{\text{PFN}}(\theta) := \mathbb{E}_{\mathcal{D}_{\text{ctx}},(\boldsymbol{x}_q,y_q)} \left[-\log q_\theta(y_q \mid \boldsymbol{x}_q, \mathcal{D}_{\text{ctx}}) \right]. \tag{2}$$

Assuming realizability and global optimization, the optimal predictor $q_{\theta^\star}$ recovers the PPD almost surely, i.e., $q_{\theta^\star}(\cdot \mid \boldsymbol{x}_q, \mathcal{D}_{\text{ctx}}) = p(\cdot \mid \boldsymbol{x}_q, \mathcal{D}_{\text{ctx}})$. For a detailed derivation, see (Müller et al., 2024).

Inference. At inference time, the pretrained TFM $q_{\theta^\star}$ receives a labeled context that represents a new task, i.e., $\mathcal{D}_{\text{ctx}} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \equiv (\boldsymbol{X}_{\text{ctx}}, \boldsymbol{y}_{\text{ctx}})$, where $\boldsymbol{X}_{\text{ctx}} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y}_{\text{ctx}} \in \mathcal{Y}^n$. Given an unlabeled query row $\boldsymbol{x}_q$, it produces the predictive distribution over the target space,

$$q_{\theta^\star}(Y_q \mid \boldsymbol{x}_q, \mathcal{D}_{\text{ctx}}), \qquad Y_q \in \mathcal{Y}. \tag{3}$$

By pretraining over diverse tasks, $q_{\theta^\star}$ is assumed to have the ability to infer the task-relevant structure from $\mathcal{D}_{\text{ctx}}$ and approximate the posterior predictive distribution of $\boldsymbol{x}_q$, thereby enabling zero-shot prediction on previously unseen tabular tasks without gradient-based parameter updates.

2.2 Tabular Foundation Models for Unsupervised TAD

TAD-specialized TFMs. Unlike general-purpose TFMs, TAD-specialized TFMs receive an unlabeled context $\mathcal{C} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ of normal data and directly estimate the anomaly probability of a query, $q_{\phi^\star}(A_q = 1 \mid \boldsymbol{x}_q, \mathcal{C})$, where $A_q \in \{0, 1\}$ represents the normal-anomalous status and $q_{\phi^\star}$ denotes the pretrained TAD-specialized TFM. To enable this inference, these models are pretrained under TAD-specific priors that generate normal contexts and normal/anomalous queries.

Shen et al. (2025) is an early TAD-specialized TFM pretrained under Gaussian mixture model priors with variance-inflated subspace anomalies. Ding et al. (2026b) extends this framework through a mixture of Gaussian-mixture, structural-causal, and copula-based priors that represent several anomaly archetypes. It further introduces a self-evolving curriculum to coordinate pretraining over heterogeneous prior families and task difficulties. Marszałek et al. (2026) similarly performs discriminative in-context anomaly detection, but explicitly supports contaminated contexts.

A notable limitation of these methods is that their notion of anomalousness is tied to the anomaly-generating mechanisms encoded in the pretraining prior; while a context can characterize dataset-specific normal structures, it cannot determine which deviations should be regarded as anomalous. Consequently, expanding anomaly coverage requires broader TAD priors together with appropriate training strategies (Ding et al., 2026b), necessitating retraining from scratch. Moreover, as general-purpose TFMs continue to advance in architecture and pretraining, these specialized approaches cannot readily inherit such improvements without substantial redesign and retraining, limiting their ability to continuously benefit from the evolving TFM ecosystem.

TAD-repurposed TFMs. Recent works have successfully repurposed pretrained general-purpose TFMs for new downstream problems (Xu et al., 2026; Hoo et al., 2026). Following this paradigm, TabPFN (Grinsztajn et al., 2026b) repurposes its predictive capabilities for unsupervised TAD by estimating sample likelihoods from its predictive distributions.[1] It factorizes the joint feature likelihood according to the chain rule. For a feature ordering $\pi = (\pi_1, \ldots, \pi_d)$, it approximates

$$\log \widehat{p}_\pi(\boldsymbol{x}_q \mid \mathcal{C}) = \sum_{j=1}^{d} \log q_{\theta^\star}\left(x_{q,\pi_j} \mid \boldsymbol{x}_{q,\pi_{<j}}, \mathcal{D}_{\mathcal{C},\pi,j}\right), \qquad \mathcal{D}_{\mathcal{C},\pi,j} = \left\{\left(\boldsymbol{x}_{i,\pi_{<j}}, x_{i,\pi_j}\right)\right\}_{i=1}^{n}, \tag{4}$$

where each feature is temporarily treated as the prediction target. Assuming that anomalous samples receive lower likelihood under the estimated joint distribution, the resulting negative log-likelihood can serve as the anomaly score for the query sample $\boldsymbol{x}_q$.

[1] https://docs.priorlabs.ai/capabilities/anomaly-detection

This extension avoids a key limitation of TAD-specialized TFMs, namely the need for substantial redesign and retraining to accommodate evolving TAD priors or improved TFM backbones, but it equates anomalousness with low joint likelihood. While this is a principled and widely used anomaly criterion, global rarity need not coincide with the application-relevant violation of normality: rare but valid observations may receive high scores. In addition, evaluating every attribute as a target across multiple feature permutations and predictor ensembles incurs a computational cost that grows with the feature dimension (Marszałek et al., 2026), as shown also in Figure 1.

3 TaskBridge: Anomaly Detection via Virtual Supervised Tasks

We introduce TaskBridge, a new framework that enables general-purpose TFMs to solve anomaly detection on arbitrary datasets without dataset-specific model tuning. The core idea is to construct *virtual supervised tasks* turning unsupervised TAD into a supervised prediction problem that these models are designed to solve.

As illustrated in Figure 2, TaskBridge constructs virtual supervised tasks that allow a pretrained TFM to perform posterior-predictive inference for each query from the virtually labeled context. These tasks are carefully designed so that the resulting anomaly score reflects the predictive support of the pretrained TFM: normal query–target pairs receive high support, while anomalous pairs tend to receive lower support. This support gap provides the basis for anomaly detection.

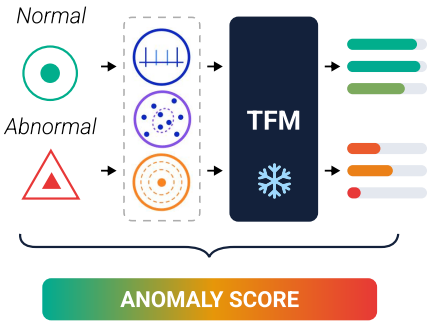


Figure 2: Overview of TaskBridge. Given a query, it assigns its virtual targets under the designed supervised tasks and distinguishes normal and anomalous queries according to the predictive supports from the pretrained TFM.

3.1 Motivation

In the supervised setting, a task can be viewed as a latent data-generating mechanism τ that induces a joint distribution $p_\tau(\mathbf{x}, Y)$. During pretraining, general-purpose TFMs repeatedly encounter supervised datasets sampled from diverse latent tasks and are optimized to approximate the corresponding PPDs for new queries by inferring task-relevant predictive structure from the labeled context, as described in Equation 1. This *in-context learning* (ICL) enables TFMs to assign higher posterior-predictive support to targets for queries that are more compatible with the inferred task.

We directly use this ICL capability of TFMs for unsupervised TAD. Specifically, we construct virtual supervised tasks that generate virtually labeled contexts from the unlabeled normal samples. Under each virtual task, the labeled normal context induces an input–target predictive structure through ICL, allowing a query to be evaluated by how well its virtual target conforms to this structure. Assuming that modern TFMs can reliably infer the predictive structures of the virtual tasks considered in this work, we design the tasks so that normal query–target pairs remain compatible with the induced structure, whereas anomalous pairs are more likely to violate it. To quantify the structural compatibility, we use the TFM's amortized posterior-predictive support for each query–target pair. Formally, let

$$\widehat{\kappa}_m(\boldsymbol{x}; \mathcal{C}) := q_{\theta^\star}(\widehat{y}_m(\boldsymbol{x}) \mid \boldsymbol{x}, \mathcal{D}_{\mathcal{C}}^m) \tag{5}$$

be the predictive support assigned by the pretrained TFM $q_{\theta^\star}$ to the virtual target $\widehat{y}_m(\boldsymbol{x})$ for input $\boldsymbol{x}$, where $\mathcal{D}_{\mathcal{C}}^m$ is the virtually labeled context induced by task m. A higher value of $\widehat{\kappa}_m(\boldsymbol{x}; \mathcal{C})$ indicates greater compatibility of $(\boldsymbol{x}, \widehat{y}_m(\boldsymbol{x}))$ with the context induced by m.

Based on this formulation, we characterize task suitability for anomaly detection through predictive support as a measure of compatibility, and identify two required conditions:

(C1) Nominal Predictive Coherence. Query–target pair $(\boldsymbol{x}, \widehat{y}_m(\boldsymbol{x}))$ generated from a normal sample $\boldsymbol{x}$ should receive consistently high predictive support from TFMs.

(C2) Selective Task Coverage. High predictive support should remain concentrated on pairs from the normal input distribution rather than extending broadly to deviant inputs. That is, $\widehat{\kappa}_m(\boldsymbol{x}; \mathcal{C})$ should be low enough if $\boldsymbol{x}$ is not sampled from the normal distribution.

3.2 NORMALITY-ANCHORED VIRTUAL SUPERVISED TASK

Following the above motivation, we construct virtual supervised tasks by extracting normality-aware structure from the unlabeled training data used as context, which is assumed to consist of normal samples. The key idea is to encode context-derived normal regularities into the virtual-target semantics, thereby inducing a task-specific predictive structure anchored to the normal distribution. We refer to such tasks as *normality-anchored virtual tasks*. From the resulting virtually labeled context, a pretrained TFM can infer this structure through ICL, under which normal query–target pairs generated by the same task rule are expected to receive high posterior-predictive support.

Definition 3.1 (Normality-anchored virtual task)**.** *Let P_0 denote the normal distribution on $\mathcal{X}$ and $\mathcal{C} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \sim P_0^n$ a clean normality-aware context. For task m, a virtual supervised task constructor $\mathfrak{T}_m = (A_m, g_m)$ induces*

$$\widehat{\psi}_{m,\mathcal{C}} = A_m(\mathcal{C}), \qquad \widehat{y}_m(\boldsymbol{x}; \mathcal{C}) = g_m\Big(\boldsymbol{x}; \widehat{\psi}_{m,\mathcal{C}}\Big), \tag{6}$$

where A_m extracts context-dependent parameters that define the supervised task, and g_m generates the virtual target for input $\boldsymbol{x}$ based on these parameters.

We call $\mathfrak{T}_m$ normality-anchored *if there exists a distribution-dependent functional a_m of P_0, with $\psi_m^0 = a_m(P_0)$, such that $\widehat{\psi}_{m,\mathcal{C}}$ consistently estimates ψ_m^0 and nontrivially determines the virtual-target semantics. Specifically, there exist sequences $\epsilon_{m,n} \to 0$ and $\delta_{m,n} \to 0$ such that*

$$\Pr_{\mathcal{C} \sim P_0^n}\Big[d_m\Big(\widehat{\psi}_{m,\mathcal{C}}, \psi_m^0\Big) > \epsilon_{m,n}\Big] \le \delta_{m,n}, \tag{7}$$

where d_m is a metric on the task-parameter space.

In Appendix A.1, we further show that normality anchoring stabilizes the induced virtual-task semantics: an independent normal query follows the same context-induced task conditional with high probability. Consequently, normal queries and their virtual-target pairs are likely to remain aligned with the predictive structure induced by the virtually labeled normality-aware context, providing the structural basis for nominal predictive coherence (C1).

Normality anchoring biases the induced predictive structure toward normal samples, encouraging TFMs to assign higher predictive support to normal query–target pairs through ICL over the virtually labeled normality-aware context. However, this alone does not always guarantee selective task coverage (C2). Depending on which normal regularities are encoded and how they are realized through the task construction, anomalous queries may still remain compatible with the induced structure, and the effectiveness of a given task can vary across datasets. We therefore construct diverse normality-anchored task candidates that capture different aspects of nominal structure under different task configurations, and perform data-adaptive task selection to retain those that best satisfy both conditions for each dataset.

3.3 NORMALITY-ANCHORED TASK FAMILIES

Rather than assuming that a single normality-anchored virtual supervised task can satisfy both conditions above across diverse tabular datasets, we construct a family of complementary virtual task templates from the given normality-aware unlabeled context $\mathcal{C}$. Each instantiated task captures a different aspect of the nominal structure, thereby inducing a distinct context-dependent predictive structure and broadening the range of structural violations that can be exposed from diverse datasets.

We first infer the dataset's attribute profile from $\mathcal{C}$. Specifically, we classify each attribute as numerical, categorical, or constant using a hybrid attribute-type inference procedure inspired by Grinsztajn et al. (2026a), and assign the dataset to a specific profile based on the relative composition of its non-constant attributes. This profile determines how the templates of task families are instantiated for the dataset. Detailed algorithms and profile-assignment criteria are provided in Appendix A.2.

Given the inferred attribute profile of the dataset, we instantiate a common set of five complementary virtual-task templates using profile-specific constructions: (i) *single-attribute*, capturing attribute-wise dependencies between a selected attribute and the remaining attributes; (ii) *localized subspace direction*, capturing dependencies within selected attribute subsets; (iii) *global direction*, capturing joint dependencies across the full attribute space; (iv) *prototype-based organization*, capturing

multimodal structure among normal samples; and (v) *distributional extremity*, capturing the relative position of a sample within the normal distribution. All families are constructed from the normality-aware context and therefore target complementary aspects of its structure. Detailed profile-specific instantiations and their correspondence to Definition 3.1 are provided in Appendix A.3.

3.4 Data-Adaptive Task Selection

Among the task families instantiated from these templates, their effectiveness for TAD depends on the characteristics of the given dataset. As illustrated in Figure 3, we therefore perform hierarchical data-adaptive task selection to retain the tasks that best satisfy the two conditions defined above.

First, for each instantiated task m corresponding to a specific template, we consider a predefined set of candidate task-construction configurations that determine the context-induced parameter $\widehat{\psi}_{m,\mathcal{C}}$. Although all candidates are normality-anchored, the configuration that best satisfies the two conditions above can vary substantially across datasets. We therefore instantiate 4–8 candidates for each task m and select the most suitable configuration in a data-adaptive manner.

To select the most suitable candidate using only the available context $\mathcal{C}$, we partition the context into a nominal held-out subset and its complementary train-side context. For split $s \in \mathcal{S}$, let $\mathcal{H}_s \subset \mathcal{C}$ denote the held-out subset and $\mathcal{C}_{-s} = \mathcal{C} \setminus \mathcal{H}_s$ the train-side context. For task m under candidate configuration $c \in \mathcal{C}_m$, let $\widehat{\psi}^c_{m,\mathcal{C}}$ denote the corresponding context-induced task parameter and define $\widehat{y}^c_m(\boldsymbol{x};\mathcal{C}) = g_m(\boldsymbol{x};\widehat{\psi}^c_{m,\mathcal{C}})$. The virtually labeled train-side context is then

$$\mathcal{D}^{m,c}_{\mathcal{C}_{-s}} := \{(\boldsymbol{x}, \widehat{y}^c_m(\boldsymbol{x};\mathcal{C})) : \boldsymbol{x} \in \mathcal{C}_{-s}\}. \tag{8}$$

Beyond the nominal held-outs, we additionally construct perturbed held-out samples calibrated from $\mathcal{C}_{-s}$,

$$\widetilde{\mathcal{H}}^{(r)}_s = T_r(\mathcal{H}_s;\mathcal{C}_{-s}), \qquad r \in \mathcal{R}_{\mathcal{C}_{-s}}, \tag{9}$$

where $\mathcal{R}_{\mathcal{C}_{-s}}$ denotes the set of structure-disrupting operators available under $\mathcal{C}_{-s}$, and T_r denotes the transformation associated with operator r.

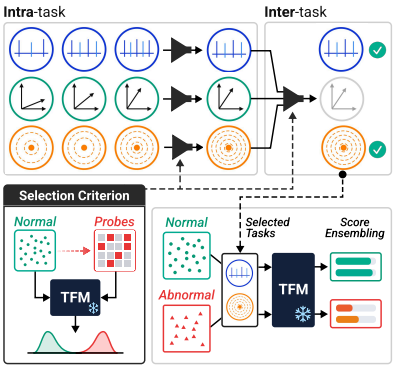


Figure 3: Overview of data-adaptive task selection. Candidates are evaluated with a pretrained TFM using nominal held-outs and structural-violation probes, and the most suitable tasks with proper configurations are retained.

These samples are deliberately constructed to depart from the normal structure characterized by $\mathcal{C}_{-s}$, thereby inducing non-trivial shifts from the normal distribution. They serve as surrogate samples for evaluating whether each candidate exhibits sufficiently selective task coverage (C2), rather than to approximate the unknown test-time anomaly distribution. Details of their construction and distributional effects are provided in Appendix A.4.

We evaluate each candidate on both the nominal held-out samples and the structural-violation probes using the pretrained general-purpose TFM $q_{\theta^\star}$, conditioned on the corresponding labeled context $\mathcal{D}^{m,c}_{\mathcal{C}_{-s}}$. For any evaluation sample $\boldsymbol{x}$ associated with split s, its posterior-predictive support is

$$\widehat{\kappa}_{m,c,s}(\boldsymbol{x}) := q_{\theta^\star}\left(\widehat{y}^c_m(\boldsymbol{x};\mathcal{C}) \mid \boldsymbol{x}, \mathcal{D}^{m,c}_{\mathcal{C}_{-s}}\right). \tag{10}$$

We aggregate these supports across held-out splits into the following sets:

$$\begin{aligned} \mathcal{K}^{\mathrm{nom}}_{m,c} &:= \{\widehat{\kappa}_{m,c,s}(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{H}_s,\ s \in \mathcal{S}\}, \\ \mathcal{K}^{\mathrm{vio}}_{m,c} &:= \left\{\widehat{\kappa}_{m,c,s}(\widetilde{\boldsymbol{x}}) : \widetilde{\boldsymbol{x}} \in \widetilde{\mathcal{H}}^{(r)}_s,\ r \in \mathcal{R}_{\mathcal{C}_{-s}},\ s \in \mathcal{S}\right\}. \end{aligned} \tag{11}$$

For each candidate c, we summarize its nominal coherence (C1) and selective task coverage (C2) by

$$\begin{aligned} \boldsymbol{r}_{m,c} &:= \left(\mathrm{Med}(\mathcal{K}^{\mathrm{nom}}_{m,c}), \mathrm{Var}(\mathcal{K}^{\mathrm{nom}}_{m,c}), \rho_{m,c}, \Delta_{m,c}\right), \\ \rho_{m,c} &:= \Pr_{u\sim\mathcal{K}^{\mathrm{nom}}_{m,c},\, v\sim\mathcal{K}^{\mathrm{vio}}_{m,c}}[u > v],\ \Delta_{m,c} := Q_{0.25}(\mathcal{K}^{\mathrm{nom}}_{m,c}) - Q_{0.75}(\mathcal{K}^{\mathrm{vio}}_{m,c}), \end{aligned} \tag{12}$$

where Med and Var denote the median and variance, respectively, Q_α denotes the α-quantile. Based on these, we perform *intra-task selection* to choose one representative configuration for each task,

$$c_m^\star = \mathrm{Select}_{\mathrm{intra}}\left(\{(c, \boldsymbol{r}_{m,c}) : c \in \mathcal{C}_m\}\right). \tag{13}$$

We then perform *inter-task selection* over the resulting task representatives to retain only the tasks most suitable for the current dataset,

$$\mathcal{M}^\star = \mathrm{Select}_{\mathrm{inter}}\left(\left\{\left(c_m^\star, \boldsymbol{r}_{m,c_m^\star}\right) : m \in \mathcal{M}\right\}\right), \tag{14}$$

where $\mathcal{M}^\star$ denotes the subset of tasks retained for the current dataset. This second stage data-adaptively retains the tasks most suitable for TAD according to the two proposed conditions. The candidate grids and details of $\mathrm{Select}_{\mathrm{intra}}$ and $\mathrm{Select}_{\mathrm{inter}}$ are provided in Appendix A.5.

3.5 Anomaly Detection

We perform anomaly detection using only the retained virtual tasks $m \in \mathcal{M}^\star$ with their selected configurations $c_m^\star$. For a query $\boldsymbol{x}_q$, the pretrained TFM evaluates the predictive support on each retained task as $\widehat{\kappa}_m(\boldsymbol{x}_q; \mathcal{C}) = q_{\theta^\star}\left(\widehat{y}_m^{c_m^\star}(\boldsymbol{x}_q; \mathcal{C}) \mid \boldsymbol{x}_q, \mathcal{D}_{\mathcal{C}}^{m,c_m^\star}\right)$. This is then converted into a task-specific anomaly score,

$$s_m(\boldsymbol{x}_q) = \varphi\left(\widehat{\kappa}_m(\boldsymbol{x}_q; \mathcal{C})\right), \tag{15}$$

where φ maps lower calibrated predictive support to stronger anomaly evidence. We aggregate task-specific anomaly scores from the retained tasks to obtain the final anomaly score:

$$S(\boldsymbol{x}_q) = \mathrm{Ensemble}\left(\{s_m(\boldsymbol{x}_q)\}_{m \in \mathcal{M}^\star}\right). \tag{16}$$

The detailed construction of φ and the ensemble strategies are provided in Appendix A.6.

4 Experiment

TaskBridge setup. We instantiate TaskBridge with the pretrained general-purpose TFM backbone TabICLv2 (Qu et al., 2026), which achieves strong predictive performance on TabArena (Erickson et al., 2025). The backbone is used with its default configuration.

To account for the predictive characteristics of each backbone, we perform a one-time backbone-specific calibration of the framework-level hyperparameters governing $\mathrm{Select}_{\mathrm{intra}}$, $\mathrm{Select}_{\mathrm{inter}}$, and $\mathrm{Ensemble}$ using a set of real-world datasets from ADBench (Han et al., 2022), which are disjoint from the main benchmark. The hyperparameters yielding the best performance on them are then fixed for each backbone before evaluation and applied unchanged to all unseen datasets. No dataset-specific fitting or further adaptation is performed. Details are provided in Appendix B.1.

Experimental Settings. All baselines, including ours, are evaluated on ODDBench (Ding et al., 2026a), which contains 790 real-world TAD datasets. For each dataset, we use the same normal-only split as the training set for conventional baselines and the context set for PFN-based methods, following Marszałek et al. (2026). Details of the dataset setup are provided in Appendix B.2.

We compare TaskBridge with 30 baselines, including 25 conventional machine learning and deep learning methods adopted from the benchmark suite of Livernoche et al. (2024) and five recent TFM-based TAD baselines. Details of the baselines and their settings are provided in Appendix B.3.

Detection performance is evaluated using both AUCROC and AUCPR on each dataset. Given the large number and heterogeneity of datasets, we complement raw performance averages with aggregate comparison metrics, such as Elo scores, to better capture relative performance across baselines. All experiments are repeated over five random seeds. Further details are provided in Appendix B.4.

4.1 Overall Performance.

As shown in Table 1 and Figure 4, TaskBridge achieves the strongest overall anomaly-detection performance among 31 methods across 790 datasets, ranking highest across all comparison metrics. In particular, the left panel of Figure 4 shows that TaskBridge remains consistently strong across dataset groups with diverse training/context resource sizes. These results demonstrate that

Table 1: Overall performance on ODDBench, showing the five methods with the highest average AUCROC among 31 baselines including TASKBRIDGE. Results are averaged over five random seeds. Parentheses indicate ranks among all 31 methods, and Total Rank is the mean rank across eight evaluation metrics; six representative metrics are shown. Best, second-best, and third-best results are highlighted in **red bold**, blue underline, and green, respectively. Our method achieves the strongest overall performance across the reported metrics. Full results are provided in Appendix C.1.

Methods	Avg. Rank (AUCROC) ↓	Avg. Rank (AUCPR) ↓	ELO (AUCROC) ↑	ELO (AUCPR) ↑	Top3 Ratio(%) (AUCROC) ↑	Top3 Ratio(%) (AUCPR) ↑
DTE-NP	$9.09_{\pm 0.03}$ (2)	$9.37_{\pm 0.02}$ (2)	$1167.7_{\pm 1.0}$ (2)	$1160.1_{\pm 0.7}$ (2)	$25.3_{\pm 0.6}$ (2)	$20.3_{\pm 0.5}$ (4)
KNN	$9.83_{\pm 0.03}$ (3)	$10.38_{\pm 0.02}$ (3)	$1144.7_{\pm 0.9}$ (3)	$1130.5_{\pm 0.7}$ (3)	$16.8_{\pm 0.4}$ (8)	$13.7_{\pm 0.6}$ (10)
TACTIC-Clean	$12.22_{\pm 0.05}$ (5)	$11.66_{\pm 0.05}$ (4)	$1086.1_{\pm 1.4}$ (6)	$1100.2_{\pm 1.3}$ (4)	$17.9_{\pm 0.4}$ (7)	$19.0_{\pm 0.5}$ (6)
OUTFORMER	$13.32_{\pm 0.05}$ (10)	$12.19_{\pm 0.02}$ (6)	$1059.1_{\pm 1.4}$ (10)	$1087.3_{\pm 0.5}$ (5)	$19.9_{\pm 0.4}$ (4)	$21.5_{\pm 0.3}$ (3)
TASKBRIDGE	**$8.39_{\pm 0.10}$ (1)**	**$7.82_{\pm 0.06}$ (1)**	**$1189.5_{\pm 3.2}$ (1)**	**$1208.1_{\pm 2.2}$ (1)**	**$44.9_{\pm 1.3}$ (1)**	**$46.3_{\pm 1.6}$ (1)**

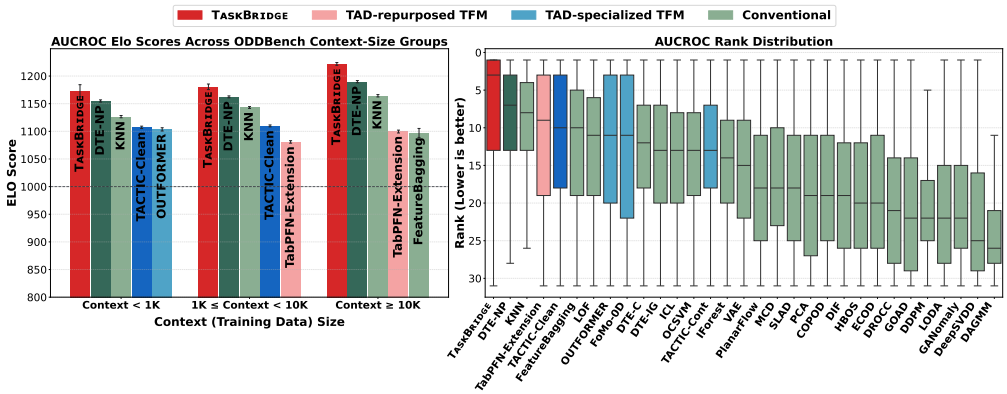


Figure 4: (Left) Elo scores across three OODBench dataset groups with different context (training data) size ($<$ 1K, 1K–10K, and $\geq$ 10K), showing the five methods with the highest average Elo scores in each group. (Right) Per-dataset AUCROC rank distributions across all baselines; colors follow Figure 1. TASKBRIDGE achieves the highest Elo score across all dataset groups and the strongest overall rank distribution. Corresponding AUCPR results are provided in Appendix C.1.

TASKBRIDGE effectively harnesses a pretrained general-purpose TFM for TAD by bridging supervised ICL and anomaly detection through data-adaptive selection of suitable normality-anchored virtual tasks, without dataset-specific tuning or TFM retraining. Full baseline/metric results for Table 1 and the corresponding AUCPR results for Figure 4 are provided in Appendix C.1.

4.2 ABLATION STUDIES

We ablate two key components of TASKBRIDGE: the normality-anchored virtual-task templates and data-adaptive task selection. As shown in Figure 5, using the full set of task templates yields the best overall performance, while variants with one or more templates removed remain competitive. This supports the complementary anomaly coverage of the task families and the effectiveness of the task-selection procedure, which can still identify informative tasks from the remaining candidates. For task selection, removing either the intra- or inter-task stage degrades performance, with a larger drop when inter-task selection is removed, highlighting the importance of retaining dataset-suitable tasks. Reducing the number of held-out splits from the default $S = 3$ to $S = 1$ largely preserves detection performance while substantially lowering task-selection cost. Further ablation details, including results for each template-removal combination, are provided in Appendix C.2.

4.3 FURTHER EXPERIMENTS

We further analyze the inference efficiency of TASKBRIDGE beyond the relative comparison in Figure 1. As detailed in Appendix C.3, task selection with the default three held-out splits takes less than one minute for most ODDBench datasets, even at context sizes near 100K, while using a single split reduces this to below 20 seconds for most datasets while largely preserving performance, as shown in Section 4.2. After task selection, inference requires only milliseconds per sample.

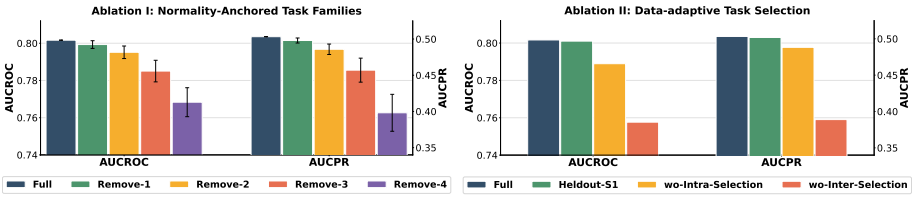


Figure 5: Ablation studies of TASKBRIDGE. (Left) Effect of the instantiated normality-anchored virtual tasks. Remove-i denotes removing i tasks from the five task templates introduced in Section 3.3; error bars indicate the variation across different removal combinations. The full task family achieves the best average performance. (Right) Effect of data-adaptive task selection. We compare the full procedure with a single held-out split, without intra-task selection, and without inter-task selection. The full selection procedure achieves the best overall performance.

We also evaluate robustness to context contamination and limited context budgets in Appendices C.4 and C.5. As shown in Figures 12 and 13, TASKBRIDGE remains competitive under contaminated contexts compared with other TFM-based approaches, and retains strong performance even when the available normality-aware context is substantially reduced, remaining competitive with baselines that use their original context/training resources. Definite performance degradation under heavier contamination also motivates improving task construction under imperfect contexts.

Finally, we assess backbone generalization by replacing the default TabICLv2 backbone with alternative pretrained TFM backbones, TabPFN v2.6 and TabPFN v3, which also show strong predictive performance on TabArena (Erickson et al., 2025), as reported in Appendix C.6. As shown in Tables 11 and 12, TASKBRIDGE with both backbones maintains strong overall performance across ODDBench among 31 TAD baselines, demonstrating that its task construction, selection, and anomaly-scoring framework generalizes beyond a single pretrained TFM backbone.

5 CONCLUSION

In this work, we introduced TASKBRIDGE, a new framework for efficiently repurposing pretrained general-purpose TFMs for unsupervised tabular anomaly detection. It bridges unsupervised TAD and supervised in-context prediction of TFMs by constructing normality-anchored virtual supervised tasks from an unlabeled normality-aware context, designed to satisfy AD-oriented conditions under which normal query–target pairs remain predictively compatible, whereas anomalous pairs tend to receive lower support from the pretrained TFM. To adapt this mechanism to each unseen dataset, TASKBRIDGE further employs data-adaptive task selection to retain the tasks with the most suitable predictive coverage. Across 790 real-world datasets, TASKBRIDGE achieves the strongest overall performance among 31 methods while preserving efficient inference with repurposed TFMs.

A limitation of our work is that TASKBRIDGE is designed to repurpose general-purpose PFN-based TFMs for TAD under the assumption that the context consists of clean normal samples. As shown in our appendix experiments, performance degrades when the context is heavily contaminated with anomalies. An important direction for future work is therefore to extend TASKBRIDGE to settings where the available context is not restricted to clean normal samples, including fully unsupervised and semi-supervised TAD. Another promising direction is to extend the framework to more structured tabular domains, such as relational databases.

AI USE STATEMENT

In this work, we used generative AI tools, including ChatGPT and Gemini, to assist with manuscript editing, research ideation and implementation, and the development of mathematical arguments and proofs. For manuscript preparation, these tools helped refine academic phrasing, correct grammatical and typographical errors, and resolve LaTeX formatting issues. For research ideation, we used them as discussion aids to explore how to operationalize the authors' core ideas and connect them to appropriate mathematical formulations and supporting arguments. For implementation, we used them to assist with generating selected portions of code and identifying potential correctness issues.

All AI-assisted content, including code, was reviewed by the authors before inclusion in the final work. Through critical examination and discussion among the authors, we assessed the suggested ideas and arguments for correctness, unsupported claims, and potential originality or attribution concerns, and verified the mathematical reasoning included in the final manuscript. Generative AI tools were not used to directly generate the figures or experimental results reported in this work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops a general framework for anomaly detection in tabular data. It does not involve human-subject studies or the collection of new personal or sensitive data; all experiments are conducted using existing benchmark datasets. The proposed method may support beneficial applications, such as identifying unusual or potentially harmful patterns in operational, financial, and security-related data. We therefore recommend that its outputs be used as decision-support signals in consequential decision-making, with appropriate human oversight and in accordance with applicable ethical and legal standards.

REPRODUCIBILITY STATEMENT

All benchmark results reported in this paper are averaged over five independent runs using random seeds 0–4. To support full reproducibility, the supplementary material includes the source code, configurations, dependency specifications, environment setup instructions, and execution scripts required to reproduce the results of TASKBRIDGE on ODDBench. It also provides the complete per-dataset results underlying the aggregate results reported in the paper. The ODDBench datasets are publicly available from the official Outformer repository at `https://github.com/psorus/Outformer`. Our implementation code is also available at `https://anonymous.4open.science/r/TaskBridge-ICLR2027-B9EF`.

REFERENCES

Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):e4150, 2021. doi: https://doi.org/10.1002/ett.4150. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150`.

Khaled Gubran Al-Hashedi and Pritheega Magalingam. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, 40:100402, 2021. ISSN 1574-0137. doi: https://doi.org/10.1016/j.cosrev.2021.100402. URL `https://www.sciencedirect.com/science/article/pii/S1574013721000423`.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pp. 93–104, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132174. doi: 10.1145/342009.335388. URL `https://doi.org/10.1145/342009.335388`.

Sihan Chen, Zhuangzhuang Qian, Wingchun Siu, Xingcan Hu, Jiaqi Li, Shawn Li, Yuehan Qin, Tiankai Yang, Zhuo Xiao, Wanghao Ye, Yichi Zhang, Yushun Dong, and Yue Zhao. Pyod 2: A python library for outlier detection with llm-powered model selection. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 2807–2810, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715196. URL https://doi.org/10.1145/3701716.3715196.

Xueying Ding, Lingxiao Zhao, and Leman Akoglu. Hyperparameter sensitivity in deep outlier detection: Analysis and a scalable hyper-ensemble solution. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9603–9616. Curran Associates, Inc., 2022. doi: 10.52202/068431-0698. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/3e9113e2bc2e700baa7d765470f140e1-Paper-Conference.pdf.

Xueying Ding, Simon Klüttermann, Haomin Wen, Yilong Chen, and Leman Akoglu. Macrodata: New benchmarks of thousands of datasets for tabular outlier detection. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 8777–8788. ACM, August 2026a. doi: 10.1145/3770855.3817520. URL http://dx.doi.org/10.1145/3770855.3817520.

Xueying Ding, Haomin Wen, Simon Klüttermann, and Leman Akoglu. From zero to hero: Advancing zero-shot foundation models for tabular outlier detection, 2026b. URL https://arxiv.org/abs/2602.03018.

Gus Eggert, Kevin Huo, Mike Biven, and Justin Waugh. Tablib: A dataset of 627m tables with context, 2023. URL https://arxiv.org/abs/2310.07875.

Arpad E Elo. The proposed uscf rating system, its development, theory, and applications. *Chess life*, 22(8):242–247, 1967.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-0519. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1697e3fb412da11dc9488249f9e7bbc9-Paper-Datasets_and_Benchmarks_Track.pdf.

Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection – a survey. *ACM Comput. Surv.*, 54(7), July 2021. ISSN 0360-0300. doi: 10.1145/3464423. URL https://doi.org/10.1145/3464423.

Mark S. Graham, Walter H.L. Pinaya, Petru-Daniel Tudosiu, Parashkev Nachev, Sebastien Ourselin, and Jorge Cardoso. Denoising diffusion models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2948–2957, June 2023.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-2.5: Advancing the state of the art in tabular foundation models, 2026a. URL https://arxiv.org/abs/2511.08667.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-3: Technical report, 2026b. URL https://arxiv.org/abs/2605.13986.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32142–32159. Curran Associates, Inc., 2022. doi: 10.52202/068431-2329. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/cf93972b116ca5268827d575f2cc226b-Paper-Datasets_and_Benchmarks.pdf.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second, 2023. URL https://arxiv.org/abs/2207.01848.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending tabpfn-v2 to time series forecasting, 2026. URL https://arxiv.org/abs/2501.02945.

David R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384 – 406, 2004. doi: 10.1214/aos/1079120141. URL https://doi.org/10.1214/aos/1079120141.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Victor Livernoche, Vineet Jain, Yashar Hezaveh, and Siamak Ravanbakhsh. On diffusion modeling for anomaly detection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 25836–25866, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6dfd16ff880a63fee9f6469fee58a496-Paper-Conference.pdf.

Martin Q. Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. The need for unsupervised outlier model selection: A review and evaluation of internal evaluation strategies. *SIGKDD Explor. Newsl.*, 25(1):19–35, July 2023. ISSN 1931-0145. doi: 10.1145/3606274.3606277. URL https://doi.org/10.1145/3606274.3606277.

Patryk Marszałek, Tomasz Kuśmierczyk, and Marek Śmieja. Tactic for navigating the unknown: Tabular anomaly detection via in-context inference, 2026. URL https://arxiv.org/abs/2603.14171.

Lukasz Maziarka, Marek Smieja, Marcin Sendera, Lukasz Struski, Jacek Tabor, and Przemyslaw Spurek. Oneflow: One-class flow for anomaly detection based on a minimal volume region. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3108223. URL http://dx.doi.org/10.1109/TPAMI.2021.3108223.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference, 2024. URL https://arxiv.org/abs/2112.10510.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model, 2026. URL https://arxiv.org/abs/2602.11139.

Yuchen Shen, Haomin Wen, and Leman Akoglu. Fomo-0d: A foundation model for zero-shot tabular outlier detection, 2025. URL https://arxiv.org/abs/2409.05672.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Jack Yi Wei and Narges Armanfard. Iclad: In-context learning for unified tabular anomaly detection across supervision regimes, 2026. URL https://arxiv.org/abs/2603.19497.

Linjie Xu, Yanlin Zhang, Quan Gan, Minjie Wang, and David Wipf. No need to train your rdb foundation model, 2026. URL https://arxiv.org/abs/2602.13697.

A METHOD DETAILS

A.1 CONSISTENCY OF NORMALITY-ANCHORED VIRTUAL TASKS

We formalize how normality anchoring stabilizes the virtual-task semantics for normal queries. Specifically, the following proposition shows that, under a local-stability condition, the task conditional induced by a finite normality-aware context agrees with its population-level counterpart with high probability.

Proposition A.1 (Consistency of the context-induced task conditional for normal queries)**.** *Under Definition 3.1, recall that $\widehat{y}_m(\boldsymbol{x};\mathcal{C}) = g_m(\boldsymbol{x};\widehat{\psi}_{m,\mathcal{C}})$. Define the deterministic task conditionals induced by the context-estimated and population-level parameters as*

$$\widehat{\pi}_{m,\mathcal{C}}(y \mid \boldsymbol{x}) := \mathbf{1}\left\{y = g_m(\boldsymbol{x};\widehat{\psi}_{m,\mathcal{C}})\right\}, \qquad \pi^0_m(y \mid \boldsymbol{x}) := \mathbf{1}\left\{y = g_m(\boldsymbol{x};\psi^0_m)\right\}. \tag{17}$$

Suppose that the virtual-task conditional is locally stable at ψ^0_m under P_0. Specifically, let

$$\omega_m(t) := \sup_{\psi:\, d_m(\psi,\psi^0_m)\le t} \Pr_{\mathbf{x}\sim P_0}\left[g_m(\mathbf{x};\psi) \neq g_m(\mathbf{x};\psi^0_m)\right], \tag{18}$$

and assume that $\omega_m(t) \to 0$ as $t \to 0$.

Then, for an independent normal query $\mathbf{x}_q \sim P_0$,

$$\Pr_{\substack{\mathcal{C}\sim P_0^n\\ \mathbf{x}_q\sim P_0}}\left[\widehat{\pi}_{m,\mathcal{C}}(\cdot \mid \mathbf{x}_q) = \pi^0_m(\cdot \mid \mathbf{x}_q)\right] = \Pr_{\substack{\mathcal{C}\sim P_0^n\\ \mathbf{x}_q\sim P_0}}\left[\widehat{y}_m(\mathbf{x}_q;\mathcal{C}) = g_m(\mathbf{x}_q;\psi^0_m)\right] \ge 1 - \omega_m(\epsilon_{m,n}) - \delta_{m,n}. \tag{19}$$

Proof. Define the successful anchoring event

$$\mathcal{E}_{m,n} := \left\{d_m\left(\widehat{\psi}_{m,\mathcal{C}}, \psi^0_m\right) \le \epsilon_{m,n}\right\}. \tag{20}$$

By Definition 3.1,

$$\Pr_{\mathcal{C}\sim P_0^n}\left[\mathcal{E}^c_{m,n}\right] \le \delta_{m,n}. \tag{21}$$

Consider any realized context $\mathcal{C}$ for which $\mathcal{E}_{m,n}$ holds. Then $d_m\left(\widehat{\psi}_{m,\mathcal{C}}, \psi^0_m\right) \le \epsilon_{m,n}$. By the definition of ω_m in Equation 18,

$$\Pr_{\mathbf{x}_q\sim P_0}\left[\widehat{y}_m(\mathbf{x}_q;\mathcal{C}) \neq g_m(\mathbf{x}_q;\psi^0_m)\right] \le \omega_m(\epsilon_{m,n}). \tag{22}$$

Let

$$\mathcal{D}_{m,n} := \left\{\widehat{y}_m(\mathbf{x}_q;\mathcal{C}) \neq g_m(\mathbf{x}_q;\psi^0_m)\right\} \tag{23}$$

denote the event that the context-induced and population-level task conditionals assign different virtual targets to the normal query.

Equation 22 holds for every realized context satisfying $\mathcal{E}_{m,n}$. Therefore, averaging the corresponding query-wise disagreement probabilities over all such context realizations gives

$$\Pr\left(\mathcal{D}_{m,n} \mid \mathcal{E}_{m,n}\right) = \mathbb{E}_{\mathcal{C}\mid\mathcal{E}_{m,n}}\left[\Pr_{\mathbf{x}_q\sim P_0}\left[\widehat{y}_m(\mathbf{x}_q;\mathcal{C}) \neq g_m(\mathbf{x}_q;\psi^0_m)\right]\right] \le \omega_m(\epsilon_{m,n}). \tag{24}$$

Here, the expectation is taken over context realizations conditioned on successful anchoring. Since each query-wise disagreement probability inside the expectation is bounded by $\omega_m(\epsilon_{m,n})$, their conditional average satisfies the same bound.

When $\mathcal{E}^c_{m,n}$ occurs, the estimated task parameter is not guaranteed to lie within $\epsilon_{m,n}$ of ψ^0_m. We therefore use the worst-case bound

$$\Pr\left(\mathcal{D}_{m,n} \mid \mathcal{E}^c_{m,n}\right) \le 1. \tag{25}$$

Applying the law of total probability over the successful and failed anchoring events yields

$$\begin{aligned}\Pr(\mathcal{D}_{m,n}) &= \Pr\left(\mathcal{D}_{m,n} \mid \mathcal{E}_{m,n}\right)\Pr(\mathcal{E}_{m,n}) \\ &\quad + \Pr\left(\mathcal{D}_{m,n} \mid \mathcal{E}^c_{m,n}\right)\Pr(\mathcal{E}^c_{m,n}) \\ &\le \omega_m(\epsilon_{m,n})\Pr(\mathcal{E}_{m,n}) + \Pr(\mathcal{E}^c_{m,n}) \\ &\le \omega_m(\epsilon_{m,n}) + \delta_{m,n}, \end{aligned} \tag{26}$$

where the final inequality follows from $\Pr(\mathcal{E}_{m,n}) \le 1$ and Equation 21.

By the deterministic definitions in Equation 17, the target-disagreement event $\mathcal{D}_{m,n}$ is equivalent to $\left\{\widehat{\pi}_{m,\mathcal{C}}(\cdot \mid \mathbf{x}_q) \neq \pi^0_m(\cdot \mid \mathbf{x}_q)\right\}$. Taking complements in Equation 26 therefore gives

$$\Pr_{\substack{\mathcal{C}\sim P_0^n \\ \mathbf{x}_q \sim P_0}}\left[\widehat{\pi}_{m,\mathcal{C}}(\cdot \mid \mathbf{x}_q) = \pi^0_m(\cdot \mid \mathbf{x}_q)\right] \ge 1 - \omega_m(\epsilon_{m,n}) - \delta_{m,n}, \tag{27}$$

which proves Equation 19. □

Thus, normality anchoring stabilizes the virtual-task semantics for normal queries: as the context-estimated task parameter approaches its population counterpart, a fresh normal query follows the same task conditional with high probability. Consequently, normal query–target pairs are likely to remain aligned with the predictive structure induced by the virtually labeled normal context, providing the structural basis for nominal predictive coherence.

A.2 Feature-Type and Data-Profile Inference

Before constructing profile-specific virtual tasks, we infer the type of each observed attribute directly from the normality-aware context. Attribute-type inference is performed before preprocessing so that the original cardinality structure of each attribute is preserved.

Let $\mathcal{C} = \{\boldsymbol{x}_i\}_{i=1}^n$ denote the context with d observed attributes. For attribute j, let n_j denote the number of non-missing observations and u_j the number of unique values among them. An attribute is regarded as constant if it contains no valid observations or if $u_j \le 1$. Otherwise, we define an adaptive cardinality threshold

$$t_j = \min\left(20,\ \max\left(2, \left\lceil\sqrt{n_j}\right\rceil\right)\right). \tag{28}$$

Attribute j is classified as categorical if

$$\left(u_j < 10 \ \vee\ u_j \le t_j\right) \ \wedge\ \left(I_j^{\text{int}} = 1 \ \vee\ u_j \le t_j\right) \ \wedge\ \frac{u_j}{n_j} \le 0.5, \tag{29}$$

where I_j^{int} is an indicator of whether the observed values of attribute j are integer-like. All remaining non-constant attributes are classified as numerical. This hybrid criterion combines cardinality, integer-likeness, and the proportion of unique values to distinguish low-cardinality categorical attributes from numerical ones.

Attribute-profile assignment. Let d_{num}, d_{cat}, and d_{const} denote the numbers of attributes inferred as numerical, categorical, and constant, respectively. Since constant attributes do not provide a meaningful type preference, we exclude them when computing the relative composition of numerical and categorical attributes. Specifically, letting $d_{\text{use}} = d_{\text{num}} + d_{\text{cat}}$, we define

$$r_{\text{num}} = \frac{d_{\text{num}}}{d_{\text{use}}}, \qquad r_{\text{cat}} = \frac{d_{\text{cat}}}{d_{\text{use}}}, \tag{30}$$

whenever $d_{\text{use}} > 0$.

The feature profile of each dataset is then determined as

$$\text{Profile}(\mathcal{C}) = \begin{cases} \texttt{numerical-only}, & d_{\text{cat}} = 0,\ d_{\text{num}} > 0, \\ \texttt{categorical-only}, & d_{\text{num}} = 0,\ d_{\text{cat}} > 0, \\ \texttt{mixed-numerical-dominant}, & d_{\text{num}}, d_{\text{cat}} > 0 \text{ and } r_{\text{num}} \ge \tau_{\text{dom}}, \\ \texttt{mixed-categorical-dominant}, & d_{\text{num}}, d_{\text{cat}} > 0 \text{ and } r_{\text{cat}} \ge \tau_{\text{dom}}, \\ \texttt{mixed}, & \text{otherwise.} \end{cases} \tag{31}$$

We use the dominance threshold $\tau_{\mathrm{dom}} = 0.80$ in all experiments. The inferred profile is used only to determine the appropriate instantiations of virtual-task templates for the observed attribute composition.

A.3 Profile-Specific Normality-Anchored Virtual-Task Instantiations

We describe the concrete instantiations of the five normality-anchored virtual-task templates introduced in Section 3.3. Let $\mathcal{J}_{\mathrm{num}}$ and $\mathcal{J}_{\mathrm{cat}}$ denote the sets of numerical and categorical attributes inferred from the context $\mathcal{C}$, respectively. Each task construction produces a discrete virtual target so that the resulting labeled context can be directly processed by a pretrained TFM backbone. While the underlying templates are shared across attribute profiles, the concrete operator is adapted to the profile of the observed attributes. For notation, we use $\mathrm{QBin}_K(z;\mathcal{C})$ to denote the assignment of a scalar statistic z to one of K classes according to empirical quantile boundaries estimated from $\mathcal{C}$.

① Single attribute: Reg2Class. This template probes attribute-wise predictive dependencies in the normal distribution by treating an individual attribute as the prediction target and the remaining attributes as predictors. We instantiate this template using ***Reg2Class***, a regression-to-classification construction motivated by the synthetic classification-task generation paradigm commonly used in general-purpose TFMs (Qu et al., 2026; Hollmann et al., 2023). Rather than directly predicting a continuous attribute, Reg2Class discretizes its values using context-derived boundaries to form a classification target.

For a numerical target attribute j, we construct

$$\widehat{y}_m(\boldsymbol{x};\mathcal{C}) = \mathrm{QBin}_K\left(x_j;\mathcal{C}\right), \tag{32}$$

where the class boundaries are empirical quantiles of attribute j in the context. Candidate numerical attributes are prioritized according to their variation.

For a categorical target attribute, the observed categories themselves define the virtual classes:

$$\widehat{y}_m(\boldsymbol{x};\mathcal{C}) = \mathrm{CatIndex}_{\mathcal{C}}(x_j), \tag{33}$$

where $\mathrm{CatIndex}_{\mathcal{C}}$ maps categories observed in the context to discrete class indices. Candidate categorical attributes are screened according to their cardinality and empirical category structure so that degenerate or excessively sparse classification tasks are avoided.

Accordingly, Reg2Class is instantiated using numerical attributes for the *numerical-only* profile and categorical attributes for the *categorical-only* profile. For the *mixed* profile, both numerical and categorical candidates are considered, with the corresponding numerical-binning or categorical-mapping rule applied according to the selected target attribute. For mixed profiles dominated by one attribute type, the dominant attribute group is preferentially used to construct the axis-aligned task.

② Localized subspace direction: Masked Projection. The second task template probes localized predictive dependencies of the normal distribution that are expressed within a subset of attributes. The underlying motivation is that an anomalous observation may violate the predictive structure of a particular subspace while remaining compatible with the normal regularities expressed by the remaining attributes. We therefore construct a scalar projection from a randomly sampled attribute subset and discretize the resulting statistic into virtual classes.

For numerical attributes, Masked Random Subspace Projection (MRSP) samples a subset $\mathcal{S}_m \subseteq \mathcal{J}_{\mathrm{num}}$ and constructs

$$z_m(\boldsymbol{x}) = \boldsymbol{w}_m^{\top} T_{\mathcal{C}}\left(\boldsymbol{x}_{\mathcal{S}_m}\right), \qquad \widehat{y}_m(\boldsymbol{x};\mathcal{C}) = \mathrm{QBin}_K\left(z_m(\boldsymbol{x});\mathcal{C}\right), \tag{34}$$

where $T_{\mathcal{C}}$ denotes the context-fitted numerical transformation and $\boldsymbol{w}_m$ is a randomly sampled projection vector that is fixed for the instantiated task.

For categorical attributes, we instead use Masked Categorical Random Projection (MCRP). Given a sampled categorical subset, each observed attribute–category token is assigned a random weight modulated by its empirical rarity in the context. For category value v of attribute j, we define the smoothed rarity statistic

$$r_j(v) = \log \frac{n + \alpha(K_j + 1)}{N_j(v) + \alpha}, \tag{35}$$

where n is the number of context samples, K_j is the number of categories observed for attribute j, $N_j(v)$ denotes the number of context samples with category value v, and $\alpha > 0$ is a smoothing constant. We then construct

$$z_m(\boldsymbol{x}) = \frac{1}{\sqrt{|\mathcal{S}_m|}} \sum_{j \in \mathcal{S}_m} r_j(x_j)\, \xi_{j,x_j}, \tag{36}$$

where $\mathcal{S}_m$ is the sampled categorical attribute subset and $\xi_{j,v}$ is a task-specific random weight. The resulting scalar score is again partitioned using context-derived quantiles QBin_K.

For mixed attributes, Masked Mixed Random Projection (MMRP) first converts numerical attributes into context-dependent quantile tokens while retaining categorical values as categorical tokens. The resulting heterogeneous token view is then processed using the same rarity-weighted random projection mechanism for categorical attributes.

Therefore, the concrete masked operator is selected according to the inferred profile. Numerical-only data use MRSP, categorical-only data use MCRP, and balanced mixed data use MMRP. For mixed-numerical-dominant data, we retain both the numerical MRSP and the MMRP; analogously, mixed-categorical-dominant data use both the MCRP and MMRP.

③ Global direction: Global Projection. This template probes global predictive dependencies of the normal distribution that are expressed across a broader set of attributes. The underlying motivation is that some anomalous observations may violate distributed multivariate relationships that are not confined to a localized subspace. Accordingly, this family captures global directional predictive structure by constructing projections over the full feature view.

For numerical-only data, Global Random Projection uses all available numerical attributes:

$$z_m(\boldsymbol{x}) = \boldsymbol{w}_m^\top T_{\mathcal{C}}(\boldsymbol{x}_{\mathcal{J}_{\mathrm{num}}}), \qquad \widehat{y}_m(\boldsymbol{x}; \mathcal{C}) = \mathrm{QBin}_K\left(z_m(\boldsymbol{x}); \mathcal{C}\right). \tag{37}$$

For categorical-only data, Global Categorical Random Projection applies the rarity-weighted categorical projection to the complete categorical feature set rather than a sampled subset.

For mixed data, Mixed Random Projection uses the complete mixed token view: numerical attributes are converted to context-derived quantile tokens and categorical attributes retain their categorical identities. A rarity-weighted random projection over all resulting tokens then defines the scalar statistic used for virtual-class construction.

In the current implementation, numerical-only and categorical-only profiles use their corresponding type-specific global projections, whereas the mixed and both mixed-dominant profiles use the Mixed Random Projection.

④ Prototype-based multimodal organization: Clustering. This template probes multimodal predictive structure of the normal distribution through context-fitted prototypes. The underlying motivation is that normal data may consist of multiple modes with distinct attribute relationships, such that an anomalous sample can become incompatible with the prototype structure induced by the context. Unlike projection-based tasks, clustering represents this structure through multiple nominal prototypes rather than a single projected ordering.

For numerical-only data, we apply K-means to the context-transformed numerical attributes and use the resulting cluster index as the virtual target:

$$\widehat{y}_m(\boldsymbol{x}; \mathcal{C}) = \arg\min_{k \in [K]} \|T_{\mathcal{C}}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_{k,\mathcal{C}}\|_2^2. \tag{38}$$

For categorical-only data, we replace Euclidean clustering with categorical K-modes. Let $\widehat{\boldsymbol{p}}_{k,\mathcal{C}}$ denote the attribute-wise modal-category prototype estimated from the context samples assigned to cluster k. The categorical discrepancy is the Hamming distance

$$d_{\mathrm{cat}}(\boldsymbol{x}, \boldsymbol{p}_k) = \sum_{j \in \mathcal{J}_{\mathrm{cat}}} \mathbf{1}[x_j \neq p_{k,j}]. \tag{39}$$

The categorical virtual target is then the index of the nearest modal prototype:

$$\widehat{y}_m(\boldsymbol{x}; \mathcal{C}) = \arg\min_{k \in [K]} d_{\mathrm{cat}}\left(\boldsymbol{x}_{\mathcal{J}_{\mathrm{cat}}}, \widehat{\boldsymbol{p}}_{k,\mathcal{C}}\right). \tag{40}$$

Thus, two samples receive the same virtual class when they are assigned to the same context-fitted categorical prototype. The prototypes are refined by alternating nearest-prototype assignments and attribute-wise modal updates.

For mixed profiles, we use a K-prototypes-style construction on a context-fitted mixed representation. First, numerical attributes are converted to context-derived quantile tokens, as in mixed random projection. Let $\widetilde{\boldsymbol{x}}_{\mathrm{num},\mathcal{C}}$ denote the resulting numerical bin-code vector, while $\boldsymbol{x}_{\mathcal{J}_{\mathrm{cat}}}$ retains the categorical values. Each mixed prototype contains the coordinate-wise median numerical bin code and the attribute-wise modal categorical value of its assigned context samples. We define

$$d_{\mathrm{num}}(\boldsymbol{x}, \boldsymbol{p}_k; \mathcal{C}) = \frac{1}{|\mathcal{J}_{\mathrm{num}}|} \sum_{j \in \mathcal{J}_{\mathrm{num}}} (\widetilde{x}_{j,\mathcal{C}} - p_{k,j})^2 \tag{41}$$

and the normalized categorical mismatch

$$\overline{d}_{\mathrm{cat}}(\boldsymbol{x}, \boldsymbol{p}_k) = \frac{1}{|\mathcal{J}_{\mathrm{cat}}|} \sum_{j \in \mathcal{J}_{\mathrm{cat}}} \mathbf{1}[x_j \neq p_{k,j}]. \tag{42}$$

The mixed prototype distance and corresponding virtual target are

$$d_{\mathrm{mix}}(\boldsymbol{x}, \boldsymbol{p}_k; \mathcal{C}) = d_{\mathrm{num}}(\boldsymbol{x}, \boldsymbol{p}_k; \mathcal{C}) + \overline{d}_{\mathrm{cat}}(\boldsymbol{x}, \boldsymbol{p}_k), \tag{43}$$

$$\widehat{y}_m(\boldsymbol{x}; \mathcal{C}) = \arg\min_{k \in [K]} d_{\mathrm{mix}}\left(\boldsymbol{x}, \widehat{\boldsymbol{p}}_{k,\mathcal{C}}; \mathcal{C}\right). \tag{44}$$

As with categorical K-modes, the prototypes are obtained by alternating assignment and prototype-update steps.

Accordingly, the numerical-only, categorical-only, and mixed profiles use K-means, categorical K-modes, and the context-discretized mixed K-prototypes construction, respectively. Both mixed-dominant profiles also use the mixed K-prototypes construction.

Distributional extremity: Radial and Rarity-based Constructions. The last fifth template probes predictive structures associated with context-relative distributional extremity. The underlying motivation is that the normality-aware context induces not only relational structure among attributes, but also a characteristic organization of samples according to their relative position within the nominal support. An anomalous sample exhibiting a scale shift or other support-relative deviation may therefore be assigned an extremity level that is incompatible with the predictive structure induced by the context.

For numerical-only data, Radial Bin computes the radial magnitude of the context-transformed numerical vector,

$$\rho(\boldsymbol{x}) = \left\| T_{\mathcal{C}}(\boldsymbol{x}) \right\|_2, \tag{45}$$

and assigns the corresponding virtual target using context-derived quantile bins: $\widehat{y}_m(\boldsymbol{x}; \mathcal{C}) = \mathrm{QBin}_K\left(\rho(\boldsymbol{x}); \mathcal{C}\right)$.

For categorical-only data, Categorical Rarity Bin estimates the smoothed empirical occurrence probability of each category from the normality-aware context. Specifically, for category value v of attribute j, we define

$$\widehat{p}_{j,\mathcal{C}}(v) = \frac{N_j(v) + \alpha}{n + \alpha(K_j + 1)}, \tag{46}$$

where n is the number of context samples, K_j is the number of categories observed for attribute j, $N_j(v)$ denotes the number of context samples taking value v at attribute j, and $\alpha > 0$ is a smoothing constant. A sample-level rarity statistic is then constructed as

$$R(\boldsymbol{x}) = -\frac{1}{|\mathcal{J}_{\mathrm{cat}}|} \sum_{j \in \mathcal{J}_{\mathrm{cat}}} \log \widehat{p}_{j,\mathcal{C}}(x_j), \tag{47}$$

where x_j is the category value of sample $\boldsymbol{x}$ at attribute j. Thus, categories that occur less frequently in the normality-aware context contribute more strongly to $R(\boldsymbol{x})$. The resulting rarity statistic is subsequently discretized using context-derived empirical quantiles to form the virtual classification target.

For mixed profiles, Mixed Categorical Rarity Bin first converts numerical attributes into context-derived quantile tokens and combines them with the original categorical tokens. The rarity score is then evaluated over this joint token representation and discretized into virtual classes. This construction is used for the balanced mixed profile as well as both mixed-dominant profiles.

Normality-anchoring characteristics of virtual tasks. Despite their different targeting predictive structures, all of the above constructions share a common source of task anchoring. Their virtual-label semantics depend on quantities estimated from the normality-aware context, including empirical quantile boundaries, category supports and frequencies, numerical transformations, and fitted prototypes. Random components such as sampled attribute subsets or projection weights determine which view of the attributes is probed, but are sampled once for the instantiation and kept fixed across context and query labeling.

Consequently, each instantiated task can be expressed in the form introduced in Equation 6. Under standard consistency of the corresponding empirical statistics, these context-dependent parameters converge to population quantities determined by P_0. Together with their nontrivial influence on the induced virtual-label semantics, the instantiated task families therefore satisfy the normality-anchoring principle of Definition 3.1.

A.4 Held-out Evaluation Sample Generation

For each dataset and random seed, we construct a common registry of held-out splits that is shared across all candidate virtual tasks. This ensures that candidate tasks are evaluated against the same normal observations. Let $\mathcal{S} := [S] = \{1, \ldots, S\}$ denote the set of held-out splits. For split $s \in \mathcal{S}$, let $\mathcal{H}_s$ denote the nominal held-out subset, $\widetilde{\mathcal{H}}_s^{(r)}$ the set of structural-violation samples generated by applying the structure-disrupting operator r, and $\mathcal{C}_{-s}$ the complementary train-side context. We set $S = 3$ as the default throughout our framework.

Nominal held-out subsets. These sets are constructed from disjoint subsets of the original normality-aware context $\mathcal{C}$. Given a hold-out ratio ρ_H, the number of held-out samples per split is

$$n_{\mathrm{H}} = \min\left\{n_{\max}, \lceil \rho_H n \rceil, \left\lfloor \frac{n}{S} \right\rfloor, n-1\right\}, \tag{48}$$

where n denotes the context size, S the number of held-out splits, and $n_{\max}$ the maximum number of held-out samples per split. In our experiments, we set $n_{\max} = 2048$, while ρ_H is selected through the backbone-specific configuration calibration described in Appendix B.1.

Context-calibrated structural-violation held-out samples. For each held-out split s, we generate operator-specific probe sets $\{\widetilde{\mathcal{H}}_s^{(r)} : r \in \mathcal{R}_{\mathcal{C}_{-s}}\}$ whose combined sample budget equals $|\mathcal{H}_s|$. The budget is distributed as evenly as possible across the active perturbation operators, such that each operator generates approximately $|\mathcal{H}_s|/|\mathcal{R}_{\mathcal{C}_{-s}}|$ samples.

The active operators in $\mathcal{R}_{\mathcal{C}_{-s}}$ are column shuffling, subset replacement, and scaled jitter. Each operator is constructed using the complementary train-side context $\mathcal{C}_{-s}$. For every perturbed sample $\widetilde{\boldsymbol{x}}$, the virtual target is computed using the same task instantiated from the full context $\mathcal{C}$. The perturbation transformation T_r associated with each operator r is defined as follows.

Column shuffling constructs synthetic rows directly from $\mathcal{C}_{-s}$. For each generated row i and attribute j, a donor row index π_{ij} is independently sampled from the rows of $\mathcal{C}_{-s}$ and

$$\widetilde{x}_{ij}^{\mathrm{shuffle}} = x_{\pi_{ij},j}. \tag{49}$$

Because the donor row is independently sampled for each attribute, the transformation approximately preserves the empirical marginal distribution of each attribute while disrupting cross-attribute dependencies.

Subset replacement starts from a sample $\boldsymbol{x}_h$ drawn from the nominal held-out subset and randomly replaces a subset of its attributes with donor values from $\mathcal{C}_{-s}$. Specifically, let $\mathcal{A}$ denote the subset of attributes selected independently with replacement probability $p_{\mathrm{rep}} = 0.3$. For each $j \in \mathcal{A}$, a donor row index ν_j is independently sampled from the rows of $\mathcal{C}_{-s}$, and the perturbed sample is defined attribute-wise as

$$\widetilde{x}_{h,j}^{\mathrm{replace}} = \begin{cases} x_{\nu_j,j}, & j \in \mathcal{A}, \\ x_{h,j}, & j \notin \mathcal{A}. \end{cases} \tag{50}$$

Thus, approximately 30% of the attributes are replaced on average, with donor values sampled independently across the selected attributes.

Scaled jitter perturbs the numerical attributes of a sample $\boldsymbol{x}_h$ drawn from the nominal held-out subset by adding feature-wise Gaussian noise scaled according to the dispersion estimated from the complementary train-side context $\mathcal{C}_{-s}$. Specifically, for each numerical attribute j,

$$\widetilde{x}_{h,j}^{\text{jitter}} = x_{h,j} + \lambda_j \widehat{d}_{j,\mathcal{C}_{-s}} \epsilon_j, \qquad \epsilon_j \sim \mathcal{N}(0,1), \tag{51}$$

where $\widehat{d}_{j,\mathcal{C}_{-s}}$ denotes the interquartile range of attribute j estimated from $\mathcal{C}_{-s}$, and we set $\lambda_j = 0.5$. Scaled jitter is omitted when no numerical attribute is available.

Distributional validity of the context-calibrated structural-violation. The preceding perturbations are used to assess the selective task coverage defined by C2 in the main paper. Rather than attempting to approximate the unknown test-time anomaly distribution, they construct controlled alternatives to the normal distribution by disrupting complementary forms of statistical regularity.

To formalize their distributional effects, consider the population idealization in which the train-side context and held-out samples are independently drawn from P_0. Let $\mathbf{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_d) \sim P_0$, let $P_{0,j}$ denote the marginal distribution of x_j, and let $P_{0,\mathcal{A}}$ denote the joint marginal distribution over an attribute subset $\mathcal{A} \subseteq [d]$. We define the total correlation of $\mathbf{x}_\mathcal{A}$ as

$$\mathrm{TC}\left(\mathbf{x}_\mathcal{A}\right) := D_{\mathrm{KL}}\left(P_{0,\mathcal{A}} \,\middle\|\, \prod_{j\in\mathcal{A}} P_{0,j}\right). \tag{52}$$

For split s, let $\widehat{P}_{j,s}$ denote the empirical marginal distribution of attribute j in $\mathcal{C}_{-s}$. Under the i.i.d. assumption, $\widehat{P}_{j,s}$ converges to $P_{0,j}$ as the train-side context grows.

The following proposition characterizes the distributions induced by these probe operators and their corresponding structural effects.

Proposition A.2 (Distributional effects of the context-calibrated probe operators)**.** *Assume that the held-out base sample is independent of $\mathcal{C}_{-s}$ and that all donor indices are sampled independently. Whenever the displayed KL divergences are well defined, the probe operators have the following effects.*

1. ***Column shuffling.*** *Conditional on $\mathcal{C}_{-s}$, column shuffling induces*

$$\widehat{Q}_s^{\text{shuffle}} = \prod_{j=1}^{d} \widehat{P}_{j,s}. \tag{53}$$

In the population limit, this becomes

$$Q^{\text{shuffle}} = \prod_{j=1}^{d} P_{0,j}, \tag{54}$$

for which

$$D_{\mathrm{KL}}\left(P_0 \,\middle\|\, Q^{\text{shuffle}}\right) = \mathrm{TC}(\mathbf{x}). \tag{55}$$

2. ***Subset replacement.*** *Conditional on a realized replacement subset $\mathcal{A} \subseteq [d]$, subset replacement induces*

$$\widehat{Q}_{\mathcal{A},s}^{\text{replace}} = P_{0,\mathcal{A}^c} \prod_{j\in\mathcal{A}} \widehat{P}_{j,s}. \tag{56}$$

In the population limit, this becomes

$$Q_\mathcal{A}^{\text{replace}} = P_{0,\mathcal{A}^c} \prod_{j\in\mathcal{A}} P_{0,j}, \tag{57}$$

and

$$\begin{aligned} D_{\mathrm{KL}}\left(P_0 \,\middle\|\, Q_\mathcal{A}^{\text{replace}}\right) = {} & I_{P_0}\left(\mathbf{x}_\mathcal{A}; \mathbf{x}_{\mathcal{A}^c}\right) \\ & + \mathrm{TC}\left(\mathbf{x}_\mathcal{A}\right). \end{aligned} \tag{58}$$

When $\mathcal{A} = \varnothing$, the operator leaves the distribution unchanged and both terms in Equation 58 are zero.

3. ***Scaled jitter.*** *Let $\mathcal{J}_{\text{num}}$ denote the set of numerical attributes and $\mathcal{J}_{\text{num}}^c$ its complement. Conditional on $\mathcal{C}_{-s}$ and the feature-wise jitter scales, scaled jitter satisfies*

$$\widetilde{\mathbf{x}}_{\mathcal{J}_{\text{num}}}^{\text{jitter}} = \mathbf{x}_{\mathcal{J}_{\text{num}}} + \boldsymbol{\eta}_s, \qquad \widetilde{\mathbf{x}}_{\mathcal{J}_{\text{num}}^c}^{\text{jitter}} = \mathbf{x}_{\mathcal{J}_{\text{num}}^c}, \tag{59}$$

where

$$\begin{aligned} \boldsymbol{\eta}_s \mid \mathcal{C}_{-s} \sim \mathcal{N}\left(\mathbf{0}, \widehat{\Sigma}_{\eta,s}\right), \qquad \boldsymbol{\eta}_s \perp\!\!\!\perp \mathbf{x} \mid \mathcal{C}_{-s}, \\ \widehat{\Sigma}_{\eta,s} := \operatorname{diag}\left(\lambda_j^2 \widehat{d}_{j,\mathcal{C}_{-s}}^2\right)_{j \in \mathcal{J}_{\text{num}}}. \end{aligned} \tag{60}$$

Consequently, whenever the corresponding second moments exist,

$$\begin{aligned} \mathbb{E}\left[\widetilde{\mathbf{x}}_{\mathcal{J}_{\text{num}}}^{\text{jitter}} \mid \mathcal{C}_{-s}\right] &= \mathbb{E}\left[\mathbf{x}_{\mathcal{J}_{\text{num}}}\right], \\ \operatorname{Cov}\left(\widetilde{\mathbf{x}}_{\mathcal{J}_{\text{num}}}^{\text{jitter}} \mid \mathcal{C}_{-s}\right) &= \operatorname{Cov}\left(\mathbf{x}_{\mathcal{J}_{\text{num}}}\right) + \widehat{\Sigma}_{\eta,s}. \end{aligned} \tag{61}$$

Hence, whenever $\widehat{\Sigma}_{\eta,s} \neq \mathbf{0}$, scaled jitter induces a nontrivial distributional shift in the numerical subvector.

Proof. For column shuffling, each attribute is sampled independently from its train-side empirical marginal, yielding Equation 53. Replacing these empirical marginals by their population counterparts gives Equation 54, and the KL identity follows directly from the definition of total correlation.

For subset replacement, the retained attributes preserve their nominal joint distribution, whereas the replaced attributes are sampled independently from their respective marginals. Hence,

$$Q_{\mathcal{A}}^{\text{replace}} = P_{0,\mathcal{A}^c} \prod_{j \in \mathcal{A}} P_{0,j}. \tag{62}$$

Assuming the corresponding densities exist, its log-density ratio with respect to P_0 decomposes as

$$\begin{aligned} \log \frac{p_0(\mathbf{x})}{p_{0,\mathcal{A}^c}(\mathbf{x}_{\mathcal{A}^c}) \prod_{j \in \mathcal{A}} p_{0,j}(\mathrm{x}_j)} &= \log \frac{p_0(\mathbf{x})}{p_{0,\mathcal{A}}(\mathbf{x}_{\mathcal{A}}) p_{0,\mathcal{A}^c}(\mathbf{x}_{\mathcal{A}^c})} \\ &\quad + \log \frac{p_{0,\mathcal{A}}(\mathbf{x}_{\mathcal{A}})}{\prod_{j \in \mathcal{A}} p_{0,j}(\mathrm{x}_j)}. \end{aligned} \tag{63}$$

Taking expectations with respect to $\mathbf{x} \sim P_0$ gives

$$\begin{aligned} D_{\text{KL}}\left(P_0 \,\middle\|\, Q_{\mathcal{A}}^{\text{replace}}\right) &= D_{\text{KL}}\left(P_0 \,\|\, P_{0,\mathcal{A}} P_{0,\mathcal{A}^c}\right) \\ &\quad + D_{\text{KL}}\left(P_{0,\mathcal{A}} \,\middle\|\, \prod_{j \in \mathcal{A}} P_{0,j}\right) \qquad (64) \\ &= I_{P_0}\left(\mathbf{x}_{\mathcal{A}}; \mathbf{x}_{\mathcal{A}^c}\right) + \text{TC}\left(\mathbf{x}_{\mathcal{A}}\right), \qquad (65) \end{aligned}$$

where the first term measures the dependence between the replaced and retained attributes, while the second measures the dependence among the replaced attributes themselves.

Finally, scaled jitter adds conditionally independent zero-mean noise to the numerical subvector. Therefore, the perturbation preserves its conditional mean, while independence eliminates the cross-covariance terms, yielding

$$\operatorname{Cov}\left(\mathbf{x}_{\mathcal{J}_{\text{num}}} + \boldsymbol{\eta}_s \mid \mathcal{C}_{-s}\right) = \operatorname{Cov}\left(\mathbf{x}_{\mathcal{J}_{\text{num}}} \mid \mathcal{C}_{-s}\right) + \widehat{\Sigma}_{\eta,s} = \operatorname{Cov}\left(\mathbf{x}_{\mathcal{J}_{\text{num}}}\right) + \widehat{\Sigma}_{\eta,s}. \tag{66}$$

Thus, jitter preserves the nominal mean while increasing dispersion along the perturbed numerical attributes. □

Proposition A.2 establishes the operator-level distributional role of the context-calibrated structural violations. Column shuffling removes global cross-attribute dependence, subset replacement disrupts both within-subset dependence and dependence between replaced and retained attributes, and scaled jitter increases dispersion along the perturbed numerical attributes. The probes therefore form

Table 2: Principal configuration axes used to construct candidate configurations for intra-task selection. Each axis is applied when the corresponding virtual-task instantiation is available for the inferred feature profile. These configurations affect only virtual-task construction; the original observed attributes are preserved for general-purpose TFM's inference.

Configuration Axis	**Applicable Feature Profiles**	**Candidates**
Number of clusters	All	$\{2, 3\}$
Numerical feature transformation	Numerical-only, all mixed variants	$\{\texttt{raw}, \texttt{robust}\}$
Number of quantile bins	All	$\{2, 3\}$
Number of quantile bins for numerical tokenization	All mixed variants	$\{2, 3\}$

complementary structural stress tests derived from the normality-aware context rather than arbitrary synthetic samples.

Nominal held-out samples assess whether the induced task maintains high predictive compatibility on normal data (C1: nominal predictive coherence), whereas structural-violation samples drawn from controlled shifted distributions serve as surrogates for assessing whether a candidate task assigns lower predictive compatibility to structural departures than to nominal held-out samples (C2: selective task coverage). Together, these provide a principled held-out criterion for selecting candidate tasks with suitable predictive coverage for tabular anomaly detection.

A.5 Details of Data-Adaptive Selection

A.5.1 Intra-task Selection

For each instantiated task m corresponding to a specific template, we construct a task-specific candidate configuration set $\mathcal{C}_m$. Each candidate $c \in \mathcal{C}_m$ determines a context-induced task parameter $\widehat{\psi}^{c}_{m,\mathcal{C}}$ and the corresponding virtual-labeling rule. These configurations control how the context-induced nominal predictive structure is converted into discrete virtual targets. The applicable configuration axes are adapted to the corresponding profile-specific task instantiation described in Appendix A.3. Table 2 summarizes the principal configuration axes used to construct the candidate sets. All candidates associated with the same dataset and random seed are evaluated using the common nominal held-out registry and structural-violation probes described in Appendix A.4.

As defined in Section 3.4, each candidate $c \in \mathcal{C}_m$ is characterized by

$$\boldsymbol{r}_{m,c} = \left(\mathrm{Med}(\mathcal{K}^{\mathrm{nom}}_{m,c}), \mathrm{Var}(\mathcal{K}^{\mathrm{nom}}_{m,c}), \rho_{m,c}, \Delta_{m,c}\right). \tag{67}$$

Intra-task selection implements $\mathrm{Select}_{\mathrm{intra}}(\cdot)$ hierarchically, so that nominal predictive coherence is first ensured before comparing the selective task coverage of candidate configurations.

Nominal-coherence filtering. We first retain candidates whose median held-out nominal compatibility meets the nominal-coherence threshold τ_{N}:

$$\mathcal{C}^{\mathrm{N}}_m = \left\{c \in \mathcal{C}_m : \mathrm{Med}(\mathcal{K}^{\mathrm{nom}}_{m,c}) \geq \tau_{\mathrm{N}}\right\}. \tag{68}$$

This step prevents a configuration with poor nominal predictive coherence from being selected solely because it strongly separates the structural-violation probes. If $\mathcal{C}^{\mathrm{N}}_m$ is empty, we directly select the candidate with the largest separation AUC as the representative configuration $c^{\star}_m$ and skip the remaining intra-task selection steps.

Task coverage-based shortlisting. Among the candidates in $\mathcal{C}^{\mathrm{N}}_m$, we next retain those with the largest separation AUC $\rho_{m,c}$. Let

$$L_m = \min\left\{\left|\mathcal{C}^{\mathrm{N}}_m\right|,\ \max\left\{L_{\min}, \left\lceil \gamma_{\mathrm{AUC}} \left|\mathcal{C}^{\mathrm{N}}_m\right| \right\rceil\right\}\right\}, \tag{69}$$

where γ_{AUC} denotes the AUC shortlist fraction and $L_{\min}$ is the minimum shortlist size whenever sufficiently many candidates are available. We then define $\mathcal{C}^{\mathrm{short}}_m$ as the L_m candidates with the largest $\rho_{m,c}$.

Representative configuration. The final representative is selected by jointly considering nominal compatibility, its stability, and conservative separation from the structural-violation probes. For candidates in $\mathcal{C}_m^{\mathrm{short}}$, let $R_{\downarrow}(\cdot)$ and $R_{\uparrow}(\cdot)$ denote normalized badness ranks for quantities for which larger and smaller values are preferred, respectively. Ties are assigned their average rank. We define

$$\begin{aligned} B_{\mathrm{intra}}(m,c) &= R_{\downarrow}\left(\mathrm{Med}(\mathcal{K}_{m,c}^{\mathrm{nom}})\right) + R_{\uparrow}\left(\mathrm{Var}(\mathcal{K}_{m,c}^{\mathrm{nom}})\right) + R_{\downarrow}\left(\Delta_{m,c}\right), \\ c_m^{\star} &= \arg\min_{c \in \mathcal{C}_m^{\mathrm{short}}} B_{\mathrm{intra}}(m,c). \end{aligned} \tag{70}$$

Importantly, the ranks in Equation 70 are recomputed within the shortlisted candidate set. Hence, $\rho_{m,c}$ acts as a separation-based qualification criterion, while the final choice balances high nominal compatibility, low nominal compatibility variance, and a large conservative quantile gap.

For the selection hyperparameters, we set $\tau_{\mathrm{N}} = 0.7$ and $L_{\min} = 2$, while γ_{AUC} is selected through the backbone-specific configuration calibration described in Appendix B.1. These choices are fixed thereafter and applied unchanged to all unseen evaluation datasets.

A.5.2 Inter-task Selection

After intra-task selection, each active virtual task $m \in \mathcal{M}$ is represented by its selected configuration $c_m^{\star}$ and the corresponding reliability statistics $\boldsymbol{r}_{m,c_m^{\star}}$. We then perform inter-task selection to remove representatives whose predictive coverage is insufficiently reliable relative to the other retained tasks. The following procedure implements $\mathrm{Select}_{\mathrm{inter}}(\cdot)$ defined in Section 3.4.

Task coverage-based prefiltering. We first require each task representative to exhibit at least chance-level separation between held-out normal samples and structural-violation probes. Specifically, we define the set of AUC-eligible tasks as

$$\mathcal{M}_{\mathrm{AUC}} = \left\{ m \in \mathcal{M} : \rho_{m,c_m^{\star}} \geq \tau_{\mathrm{A}} \right\}, \tag{71}$$

where we set $\tau_{\mathrm{A}} = 0.50$. If $\mathcal{M}_{\mathrm{AUC}}$ is empty, we replace it with the singleton set containing the task with the largest $\rho_{m,c_m^{\star}}$, ensuring that at least one task remains available for scoring.

Inter-task ranking. For the AUC-eligible tasks, we jointly compare nominal predictive coherence, its stability, and selective task coverage from the structural-violation probes. Let $R_{\downarrow}(\cdot)$ and $R_{\uparrow}(\cdot)$ denote normalized badness ranks for quantities for which larger and smaller values are preferred, respectively. The ranks are recomputed over $\mathcal{M}_{\mathrm{AUC}}$, with ties assigned their average rank. We define

$$\begin{aligned} B_{\mathrm{inter}}(m) = w_A R_{\downarrow}\left(\rho_{m,c_m^{\star}}\right) &+ R_{\downarrow}\left(\mathrm{Med}(\mathcal{K}_{m,c_m^{\star}}^{\mathrm{nom}})\right) \\ &+ R_{\uparrow}\left(\mathrm{Var}(\mathcal{K}_{m,c_m^{\star}}^{\mathrm{nom}})\right) + R_{\downarrow}\left(\Delta_{m,c_m^{\star}}\right), \qquad m \in \mathcal{M}_{\mathrm{AUC}}, \end{aligned} \tag{72}$$

where w_A controls the relative contribution of the AUC-based separation criterion. A smaller $B_{\mathrm{inter}}(m)$ indicates a more reliable task representative.

Based on $B_{\mathrm{inter}}(m)$, the final task set is obtained by applying an inter-task retention policy π_{ret} to the AUC-eligible task set:

$$\mathcal{M}^{\star} = \pi_{\mathrm{ret}}\left(\mathcal{M}_{\mathrm{AUC}}; \{B_{\mathrm{inter}}(m)\}_{m \in \mathcal{M}_{\mathrm{AUC}}}\right), \tag{73}$$

where π_{ret} determines how the relative reliability of the AUC-eligible task representatives is translated into the final retained set.

Both the AUC-rank weight w_A and the retention policy π_{ret} are selected through the backbone-specific configuration calibration using the real-world development datasets described in Appendix B.1. Once selected, they are fixed and applied unchanged to all unseen evaluation datasets.

A.6 Task-wise Anomaly Scoring and Score Aggregation

We describe how the predictive support of each retained virtual supervised task is converted into anomaly evidence and subsequently aggregated across the surviving tasks. We consider the specific realization of the task-wise scoring function φ together with several ensemble operators. The ensemble strategy is selected through backbone-specific configuration calibration using the real-world development datasets described in Appendix B.1.

Task-wise anomaly scoring. We consider *Held-out Calibrated Surprisal* as the principal form of anomaly evidence for φ. For each retained task $m \in \mathcal{M}^\star$, let $\widehat{\kappa}_m(\boldsymbol{x};\mathcal{C})$ denote the predictive support assigned by the pretrained TFM, and let $\mathcal{K}_m^{\mathrm{nom}} := \mathcal{K}_{m,c_m^\star}^{\mathrm{nom}}$ denote the held-out nominal support collection associated with its selected configuration $c_m^\star$.

We calibrate the query support against the held-out nominal support distribution $\mathcal{K}_m^{\mathrm{nom}}$ using

$$p_m(\boldsymbol{x}) = \frac{1 + \sum_{q \in \mathcal{K}_m^{\mathrm{nom}}} \mathbf{1}\left[q \le \widehat{\kappa}_m(\boldsymbol{x};\mathcal{C})\right]}{|\mathcal{K}_m^{\mathrm{nom}}| + 1}, \tag{74}$$

and define

$$s_m(\boldsymbol{x}) = -\log(p_m(\boldsymbol{x}))\,. \tag{75}$$

Task-score aggregation (ensemble). We first consider direct aggregation of the task-wise scores $s_m(\boldsymbol{x})$ using either min or low2mean aggregation. The latter is defined as

$$S_{\mathrm{low2}}(\boldsymbol{x}) = \frac{1}{|\mathcal{L}_2(\boldsymbol{x})|} \sum_{m^\star \in \mathcal{L}_2(\boldsymbol{x})} s_{m^\star}(\boldsymbol{x}), \tag{76}$$

where $\mathcal{L}_2(\boldsymbol{x})$ contains the $\min(2, |\mathcal{M}^\star|)$ surviving tasks with the smallest anomaly scores. These operators provide conservative consensus criteria, with S_{low2} offering a relaxed alternative to min aggregation.

For adaptive aggregation, we compute task-suitability scores using only the surviving tasks. Let $K = |\mathcal{M}^\star|$ and define

$$B_{\mathrm{agg}}(m) = \frac{w_A r_A(m) + r_M(m) + r_V(m) + r_G(m)}{K}, \tag{77}$$

where r_A, r_M, r_V, and r_G are the ordinal badness ranks of separation AUC, nominal median support, nominal support variance, and quantile gap, respectively, recomputed over the surviving tasks using the same preference directions as in inter-task selection. A smaller $B_{\mathrm{agg}}(m)$ indicates greater task suitability for anomaly detection. We consider the following adaptive strategies.

- **Reliability-spread adaptive aggregation.** We compute

$$\Delta_{\mathrm{rel}} = \max_{m \in \mathcal{M}^\star} B_{\mathrm{agg}}(m) - \min_{m \in \mathcal{M}^\star} B_{\mathrm{agg}}(m). \tag{78}$$

  When $K \ge 3$ and $\Delta_{\mathrm{rel}} < 1.0$, the task reliabilities are regarded as sufficiently similar, and a consensus-oriented operator (min or low2mean) is used. Otherwise, aggregation favors the most suitable task(s) according to B_{agg}.

- **Reliability-weighted aggregation.** Let $\bar{r}_m$ denote the normalized rank derived from $B_{\mathrm{agg}}(m)$. We define

$$\omega_m = \frac{(\bar{r}_m + \epsilon_w)^{-1}}{\sum_{j \in \mathcal{M}^\star} (\bar{r}_j + \epsilon_w)^{-1}}, \qquad S_{\mathrm{RW}}(\boldsymbol{x}) = \sum_{m \in \mathcal{M}^\star} \omega_m s_m(\boldsymbol{x}), \tag{79}$$

  where $\epsilon_w = 0.1$ prevents the weight assigned to the highest-ranked task from becoming singular.

B Experimental Details

B.1 Details of TaskBridge Setup

TaskBridge employs hierarchical task selection through $\mathrm{Select}_{\mathrm{intra}}$ and $\mathrm{Select}_{\mathrm{inter}}$ to data-adaptively retain suitable virtual tasks and their normality-anchored configurations, followed by $\mathrm{Ensemble}$ to aggregate their task-wise anomaly scores.

These components depend on the predictive characteristics of the underlying pretrained general-purpose TFM. In particular, task selection relies on the predictive support assigned to held-out samples, while score aggregation accounts for the relative predictive reliability of the backbone TFM across the retained tasks. We therefore perform a one-time backbone-specific calibration of the corresponding framework-level hyperparameters, as described in Section 4. Once selected, these hyperparameters are fixed for each backbone and applied unchanged to all unseen datasets.

Table 3: Statistics of the ADBench datasets.

Dataset	#Samples	#Feat.	#Anom.	%Anom.	Category
ALOI	49534	27	1508	3.04	Image
annthyroid	7200	6	534	7.42	Healthcare
backdoor	95329	196	2329	2.44	Network
breastw	683	9	239	34.99	Healthcare
campaign	41188	62	4640	11.27	Finance
cardio	1831	21	176	9.61	Healthcare
Cardiotocography	2114	21	466	22.04	Healthcare
celeba	202599	39	4547	2.24	Image
cover	286048	10	2747	0.96	Botany
donors	619326	10	36710	5.93	Sociology
fault	1941	27	673	34.67	Physical
fraud	284807	29	492	0.17	Finance
glass	214	7	9	4.21	Forensic
Hepatitis	80	19	13	16.25	Healthcare
http	567498	3	2211	0.39	Web
InternetAds	1966	1555	368	18.72	Image
Ionosphere	351	33	126	35.90	Oryctognosy
landsat	6435	36	1333	20.71	Astronautics
letter	1600	32	100	6.25	Image
Lymphography	148	18	6	4.05	Healthcare
magic.gamma	19020	10	6688	35.16	Physical
mammography	11183	6	260	2.32	Healthcare
mnist	7603	100	700	9.21	Image
musk	3062	166	97	3.17	Chemistry
optdigits	5216	64	150	2.88	Image
PageBlocks	5393	10	510	9.46	Document
pendigits	6870	16	156	2.27	Image
Pima	768	8	268	34.90	Healthcare
satellite	6435	36	2036	31.64	Astronautics
satimage-2	5803	36	71	1.22	Astronautics
shuttle	49097	9	3511	7.15	Astronautics
skin	245057	3	50859	20.75	Image
smtp	95156	3	30	0.03	Web
SpamBase	4207	57	1679	39.91	Document
speech	3686	400	61	1.65	Linguistics
Stamps	340	9	31	9.12	Document
thyroid	3772	6	93	2.47	Healthcare
vertebral	240	6	30	12.50	Biology
vowels	1456	12	50	3.43	Linguistics
Waveform	3443	21	100	2.90	Physics
WBC	223	9	10	4.48	Healthcare
WDBC	367	30	10	2.72	Healthcare
Wilt	4819	5	257	5.33	Botany
wine	129	13	10	7.75	Chemistry
WPBC	198	33	47	23.74	Healthcare
yeast	1484	8	507	34.16	Biology

ADBench. For backbone-specific calibration, we use 46 real-world datasets from ADBench (Han et al., 2022). Table 3 summarizes their dataset statistics. For each dataset, we construct the training/context split from normal samples only. Specifically, when the anomaly ratio exceeds 30%, we use 50% of the available normal samples as the training/context set; otherwise, we use 60%. For datasets containing more than $100{,}000$ total samples, we apply the without-replacement subsampling procedure.

Searched framework-level hyperparameters. The framework-level hyperparameters governing $\mathrm{Select}_{\mathrm{intra}}$, $\mathrm{Select}_{\mathrm{inter}}$, and $\mathrm{Ensemble}$ are calibrated to the predictive characteristics of each pretrained backbone TFM. The backbone itself is kept at the default configuration provided by its original paper and repository. The principal search dimensions are summarized below.

For the hierarchical task-selection procedure:

- **Held-out fraction ρ_H:** $\{0.10, 0.15, 0.20\}$.
- **Intra-family separation AUC shortlist fraction γ_{AUC}:** $\{0.50, 0.60, 0.70\}$.
- **Inter-family separation AUC-rank weight w_A:** $\{1.5, 2.0\}$.
- **Inter-family retention policy π_{ret}:** $\{\textsc{All}, \textsc{Best}, \textsc{Top}\text{-}k\}$.

For inter-family retention, ALL retains all AUC-eligible task representatives in $\mathcal{M}_{\mathrm{AUC}}$, whereas BEST retains only the task with the smallest B_{inter}. TOP-k retains the top $k = \lceil 0.5\,|\mathcal{M}_{\mathrm{AUC}}| \rceil$ tasks according to B_{inter}.

For task ensembling, we search over the candidate realizations of $\mathrm{Ensemble}(\cdot)$ defined in Appendix A.6. The combination of task-selection and aggregation configurations yielding the best average performance on the ADBench datasets is selected for each backbone as its calibrated framework-level configuration and then fixed for all unseen evaluation datasets.

B.2 Details of the Evaluation Datasets

Implementation Setting. All experiments were conducted on a machine equipped with NVIDIA H200 GPUs and two Intel Xeon Platinum 8462Y+ CPUs, providing 64 CPU cores (128 logical threads) in total.

ODDBench. ODDBench is a large-scale, real-world benchmark designed for tabular outlier/anomaly detection. Carefully curated from the large-scale TabLib repository (Eggert et al., 2023), the benchmark comprises 790 real-world datasets with semantically meaningful anomalies, such as fraud, failures, defects, and attacks, across diverse application domains. These datasets are partitioned into 690 publicly available sets and 100 private sets.

It further provides standardized train/test splits under a one-class setting, where the training set contains only normal samples and the test set contains the remaining normal samples together with all anomalous samples. Its substantially larger scale and broader diversity than conventional anomaly detection benchmarks provide a more comprehensive and statistically robust testbed for evaluating unsupervised tabular anomaly detection baselines. Further details are provided in the original ODDBench paper (Ding et al., 2026a).

Preprocessing. Following prior tabular anomaly-detection studies (Shen et al., 2025; Livernoche et al., 2024), we apply feature-wise z-score normalization to the input data used by conventional baselines. For TAD-specialized TFM baselines, we follow the preprocessing pipeline required by each released model and its corresponding input architecture. Likewise, TabPFN-Extension and TASKBRIDGE use the native input preprocessing of their underlying TFM backbones.

Subsampling of large datasets. To ensure a consistent computational budget across datasets, we limit the total number of samples in each dataset to at most 100,000. Specifically, if

$$n_{\mathrm{train}} + n_{\mathrm{test}} > 100{,}000, \tag{80}$$

where n_{train} and n_{test} denote the numbers of samples in the training/context split and test split, respectively, we perform without-replacement subsampling while approximately preserving the original train–test ratio. Let $N_{\max} = 100{,}000$. The retained split sizes are determined as

$$n'_{\mathrm{train}} = \mathrm{round}\left(N_{\max}\frac{n_{\mathrm{train}}}{n_{\mathrm{train}} + n_{\mathrm{test}}}\right), \qquad n'_{\mathrm{test}} = N_{\max} - n'_{\mathrm{train}}. \tag{81}$$

The training/context subset is then sampled uniformly without replacement using a seed-specific random generator. Since the ODDBench training split contains only normal samples, no stratification is required for this split. For the test set, we perform stratified random subsampling to preserve the original proportion of normal and anomalous samples as closely as possible. This procedure preserves the overall train–test structure and test contamination ratio while maintaining a tractable computational cost.

B.3 BASELINE SETTINGS

Conventional baselines. Following (Shen et al., 2025; Ding et al., 2026b), we compare TASKBRIDGE against 25 widely used unsupervised anomaly and outlier detection methods adopted from DTE (Livernoche et al., 2024). The classical and shallow baselines include ECOD, iForest, kNN, LOF, OCSVM, PCA, MCD, HBOS, COPOD, FeatureBagging, and LODA. These methods are implemented using PyOD (Chen et al., 2025), a widely used Python library for outlier detection. For deep learning-based baselines, we include VAE, Deep-SVDD, SLAD, ICL, GOAD, DDPM, PlanarFlow, DAGMM, DROCC, GANomaly, and DIF. We additionally evaluate the diffusion-based DTE variants, including DTE-C, DTE-IG, and DTE-NP.

Consistent with the evaluation protocols of prior TFM-based anomaly detection studies (Shen et al., 2025; Ding et al., 2026b), we use the default PyOD hyperparameter configurations for all classical and shallow baselines. For VAE and Deep-SVDD, we likewise use the implementations and default configurations provided by PyOD. For the remaining methods, we follow the hyperparameter settings reported in their original papers and the official DTE repository (Livernoche et al., 2024).

PFN-based baselines. We include five TFM-based baselines for tabular anomaly detection: FoMo-0D (Shen et al., 2025), OUTFORMER (Ding et al., 2026b), TACTIC-CLEAN and TACTIC-CONT (Marszałek et al., 2026), and the official TabPFN anomaly-detection extension[2]. TACTIC-CLEAN and TACTIC-CONT correspond to independently pretrained variants designed for clean and contaminated contexts, respectively. For the TabPFN anomaly-detection extension, we use the latest TabPFN v3.0 backbone available in our evaluation setup.

Following the official implementations and experimental settings described in the corresponding papers, we configure each baseline as follows:

For FoMo-0D, we use the officially released 100-dimensional variant. For datasets with fewer than 100 attributes, the inputs are rescaled and zero-padded to 100 dimensions; for higher-dimensional datasets, 100 attributes are randomly subsampled. For OUTFORMER, which follows the same 100-dimensional input setting as FoMo-0D, we use the officially released model without ensembling. We apply a quantile transformation to the input features and limit the context size to at most 5,000 samples for both baselines, following the setup described in their papers and code repositories.

For TACTIC-CLEAN and TACTIC-CONT, following the preprocessing protocol described in the official paper and repository, we set the input dimensionality to 50 attributes and apply MinMax scaling to the input features. Datasets with fewer than 50 attributes are zero-padded, whereas datasets with more than 50 attributes are evaluated using an ensemble of ten randomly sampled 50-feature subsets. We do not impose an additional context-length limit for either TACTIC variant.

For the TabPFN anomaly-detection extension, we do not impose a context-size limit. We use the default hyperparameters and configurations of the naive classifier- and regressor-based anomaly detection schemes provided in the official TabPFN-Extension implementation.

To ensure a focused and fair comparison, we restrict the tabular foundation model baselines to PFN-based approaches, as our work specifically targets anomaly detection through the predictive interface of PFN-based TFMs. We therefore exclude foundation models built on fundamentally different pretraining or inference paradigms, since their inclusion would confound the comparison of PFN-specific design choices. We further require baselines to provide official code, pretrained checkpoints, and sufficiently specified inference configurations for faithful reproduction. Accordingly, we exclude ICLAD (Wei & Armanfard, 2026), which releases pretrained checkpoints on Hugging Face but does not provide the official inference pipeline or complete inference settings.

Our setting. For TASKBRIDGE, we fix all configurations of the underlying general-purpose TFM backbone, TabICLv2. We use the pretrained checkpoint released through the official TabICLv2 Hugging Face repository[3] and its default classifier configuration specified in the official repository. Following the one-time real-world configuration search described in Appendix B.1, we fix the task-selection and score-aggregation configurations for the TabICLv2 backbone as follows:

$$\{\rho_H = 0.10,\ \gamma_{\mathrm{AUC}} = 0.70,\ w_A = 1.5,\ \pi_{\mathrm{ret}} = \textsc{All},\ \mathrm{Ensemble} = \textsc{low2mean}\}\,.$$

[2] https://docs.priorlabs.ai/capabilities/anomaly-detection

[3] https://huggingface.co/papers/2602.11139

These configurations are then held fixed and applied to all unseen datasets without dataset-specific tuning. Other fixed framework configurations are specified in Appendices A.4, A.5, A.6 and B.1.

B.4 Evaluation Metrics

Based on the per-dataset AUCROC and AUCPR results, we evaluate the relative performance of each method using several aggregate metrics beyond the average score and average rank. This provides a more robust comparison across the large and heterogeneous set of evaluation datasets. All aggregate metrics are computed independently for AUCROC and AUCPR.

Elo score. We use an Elo-style rating (Elo, 1967) to summarize pairwise performance across datasets. For each dataset, every pair of methods is compared according to its detection performance, assigning one point for a win, one-half point for a tie, and zero points for a loss. We estimate the relative strength β_i of each method i from these pairwise outcomes using the Bradley–Terry model (Hunter, 2004) and map the resulting strength to the Elo scale as

$$R_i = 1000 + 400 \log_{10} \beta_i. \tag{82}$$

This parameterization follows the standard Elo interpretation, under which the expected pairwise score of method i against method j is

$$E_{ij} = \frac{1}{1 + 10^{(R_j - R_i)/400}}. \tag{83}$$

A higher Elo score therefore indicates stronger and more consistent pairwise performance across the evaluation datasets.

Top-3 ratio. The Top-3 ratio measures how frequently a method ranks among the three best-performing methods across datasets. Specifically, we rank all methods within each dataset according to the corresponding evaluation metric and compute

$$\mathrm{Top3}(i) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[r_{i,n} \leq 3\right], \tag{84}$$

where $r_{i,n}$ denotes the rank of method i on dataset n, and N is the total number of datasets. A larger Top-3 ratio indicates that a method more frequently achieves competitive performance across datasets.

Rank distribution. As shown in Figure 4 and Figure 6, we further report the distribution of per-dataset ranks for each method. For each dataset, methods are ranked according to their detection performance, and the resulting ranks are collected across all 790 ODDBench datasets. We visualize these distributions using box plots, which provide a complementary view of both the typical rank and its variability across heterogeneous datasets.

C Further Experiments

C.1 Extended Overall Performance Analysis

This section provides an extended evaluation of TaskBridge against a diverse set of tabular anomaly detection baselines across 790 ODDBench datasets, complementing the results reported in the main paper (Section 4.1).

Table 4, extending Table 1 in the main paper, compares TaskBridge with 31 baselines using eight complementary evaluation metrics: average performance, average rank, Elo score, and Top3-ratio, each computed for AUCROC and AUCPR. Details of the Elo score and Top3-ratio are provided in Appendix B.4. Per-dataset results for all methods are provided in the supplementary material due to the large number of evaluation datasets and baselines.

During evaluation, TACTIC, TabPFN-Extension, and VAE fail to produce results on a small subset of datasets due primarily to out-of-memory errors or dataset–model incompatibilities. To maintain a common set of datasets for aggregate comparison, these missing results are imputed using the average performance of the remaining successfully evaluated methods on the corresponding dataset.

Table 4: Extended results corresponding to Table 1. Total Rank is computed as the mean rank among 31 methods across eight evaluation metrics: average AUCROC and AUCPR, average AUCROC and AUCPR ranks, Elo scores for AUCROC and AUCPR, and Top-3 Ratios for AUCROC and AUCPR. Our method achieves the strongest overall performance across these metrics.

	Methods	Avg. AUCROC ↑	Avg. Rank (AUCROC) ↓	Avg. AUCPR ↑	Avg. Rank (AUCPR) ↓	ELO (AUCROC) ↑	ELO (AUCPR) ↑	Top-3 Ratio (AUCROC) ↑	Top-3 Ratio (AUCPR) ↑	Total Rank ↓
Conventional	DTE-NP	$76.96_{\pm 0.01}$ (2)	$9.09_{\pm 0.03}$ (2)	$43.63_{\pm 0.01}$ (2)	$9.37_{\pm 0.02}$ (2)	$1167.7_{\pm 1.0}$ (2)	$1160.1_{\pm 0.7}$ (2)	$25.3_{\pm 0.6}$ (2)	$20.3_{\pm 0.5}$ (4)	2.25 (2)
	KNN	$76.25_{\pm 0.01}$ (3)	$9.83_{\pm 0.03}$ (3)	$42.41_{\pm 0.01}$ (5)	$10.38_{\pm 0.02}$ (3)	$1144.7_{\pm 0.9}$ (3)	$1130.5_{\pm 0.7}$ (3)	$16.8_{\pm 0.4}$ (8)	$13.7_{\pm 0.6}$ (10)	4.75 (3)
	FeatureBagging	$75.59_{\pm 0.16}$ (4)	$12.14_{\pm 0.20}$ (4)	$39.30_{\pm 0.56}$ (10)	$13.19_{\pm 0.21}$ (10)	$1086.9_{\pm 4.8}$ (4)	$1061.4_{\pm 5.2}$ (10)	$19.0_{\pm 1.1}$ (6)	$18.0_{\pm 1.1}$ (8)	7.00 (6)
	LOF	$74.17_{\pm 0.02}$ (6)	$12.83_{\pm 0.03}$ (7)	$39.97_{\pm 0.02}$ (9)	$12.97_{\pm 0.04}$ (8)	$1068.9_{\pm 0.9}$ (7)	$1066.0_{\pm 1.1}$ (8)	$13.1_{\pm 0.4}$ (10)	$14.3_{\pm 0.5}$ (9)	8.00 (9)
	DTE-C	$73.46_{\pm 0.05}$ (9)	$13.02_{\pm 0.11}$ (8)	$38.16_{\pm 0.09}$ (11)	$13.75_{\pm 0.15}$ (12)	$1063.9_{\pm 3.0}$ (8)	$1046.7_{\pm 3.6}$ (12)	$10.9_{\pm 0.6}$ (13)	$9.4_{\pm 0.4}$ (15)	11.00 (11)
	DTE-IG	$73.34_{\pm 0.43}$ (11)	$13.09_{\pm 0.30}$ (9)	$41.92_{\pm 0.34}$ (6)	$12.15_{\pm 0.23}$ (5)	$1061.4_{\pm 7.0}$ (9)	$1085.1_{\pm 5.6}$ (6)	$15.5_{\pm 1.3}$ (9)	$18.5_{\pm 0.8}$ (7)	7.75 (7)
	OCSVM	$71.88_{\pm 0.01}$ (13)	$13.88_{\pm 0.02}$ (12)	$37.18_{\pm 0.01}$ (12)	$14.03_{\pm 0.04}$ (13)	$1042.9_{\pm 0.7}$ (12)	$1039.9_{\pm 1.1}$ (13)	$8.5_{\pm 0.4}$ (17)	$7.2_{\pm 0.2}$ (18)	13.75 (14)
	ICL	$71.82_{\pm 0.10}$ (14)	$14.02_{\pm 0.08}$ (13)	$40.07_{\pm 0.15}$ (8)	$13.11_{\pm 0.15}$ (9)	$1040.8_{\pm 1.6}$ (14)	$1063.3_{\pm 3.5}$ (9)	$11.5_{\pm 0.3}$ (11)	$12.7_{\pm 0.4}$ (11)	11.12 (12)
	IForest	$71.26_{\pm 0.05}$ (15)	$15.07_{\pm 0.05}$ (15)	$31.26_{\pm 0.10}$ (20)	$16.53_{\pm 0.10}$ (16)	$1021.1_{\pm 1.4}$ (15)	$986.2_{\pm 2.4}$ (16)	$4.7_{\pm 0.4}$ (27)	$4.4_{\pm 0.2}$ (27)	18.88 (18)
	MCD	$69.10_{\pm 0.19}$ (17)	$16.55_{\pm 0.15}$ (17)	$31.10_{\pm 0.22}$ (21)	$17.88_{\pm 0.12}$ (19)	$982.3_{\pm 3.4}$ (17)	$950.2_{\pm 2.7}$ (19)	$11.0_{\pm 0.6}$ (12)	$10.4_{\pm 0.4}$ (12)	16.75 (16)
	VAE	$70.20_{\pm 0.04}$ (16)	$15.76_{\pm 0.06}$ (16)	$35.34_{\pm 0.06}$ (15)	$15.99_{\pm 0.10}$ (15)	$998.7_{\pm 1.5}$ (16)	$993.8_{\pm 2.2}$ (15)	$7.6_{\pm 0.3}$ (19)	$6.4_{\pm 0.6}$ (22)	16.75 (16)
	PlanarFlow	$68.11_{\pm 0.40}$ (18)	$17.11_{\pm 0.13}$ (18)	$32.95_{\pm 0.17}$ (17)	$17.53_{\pm 0.04}$ (17)	$968.3_{\pm 3.5}$ (18)	$958.6_{\pm 1.3}$ (17)	$10.8_{\pm 0.7}$ (14)	$10.1_{\pm 0.6}$ (13)	16.50 (15)
	SLAD	$66.87_{\pm 0.05}$ (21)	$17.84_{\pm 0.03}$ (19)	$33.49_{\pm 0.09}$ (16)	$17.59_{\pm 0.03}$ (18)	$948.9_{\pm 0.6}$ (19)	$955.7_{\pm 0.7}$ (18)	$6.6_{\pm 0.2}$ (22)	$7.1_{\pm 0.1}$ (20)	19.12 (19)
	COPOD	$67.20_{\pm 0.01}$ (20)	$18.37_{\pm 0.04}$ (20)	$27.99_{\pm 0.01}$ (29)	$19.51_{\pm 0.06}$ (28)	$941.2_{\pm 1.1}$ (20)	$913.2_{\pm 1.5}$ (27)	$3.9_{\pm 0.3}$ (28)	$4.2_{\pm 0.2}$ (28)	25.00 (25)
	PCA	$64.50_{\pm 0.03}$ (26)	$18.49_{\pm 0.03}$ (21)	$29.14_{\pm 0.02}$ (26)	$18.68_{\pm 0.03}$ (22)	$936.2_{\pm 0.7}$ (22)	$931.7_{\pm 0.7}$ (22)	$5.9_{\pm 0.1}$ (24)	$4.7_{\pm 0.2}$ (26)	23.62 (23)
	ECOD	$68.10_{\pm 0.01}$ (19)	$18.52_{\pm 0.06}$ (22)	$28.25_{\pm 0.02}$ (28)	$19.01_{\pm 0.08}$ (23)	$938.2_{\pm 1.4}$ (21)	$925.9_{\pm 1.8}$ (23)	$6.4_{\pm 0.3}$ (23)	$5.9_{\pm 0.3}$ (23)	22.75 (22)
	HBOS	$66.58_{\pm 0.02}$ (22)	$18.62_{\pm 0.06}$ (23)	$29.90_{\pm 0.03}$ (24)	$18.34_{\pm 0.05}$ (20)	$934.3_{\pm 1.5}$ (23)	$941.5_{\pm 1.2}$ (20)	$6.9_{\pm 0.3}$ (21)	$7.1_{\pm 0.4}$ (19)	21.50 (21)
	DIF	$65.16_{\pm 0.02}$ (24)	$18.87_{\pm 0.04}$ (25)	$28.53_{\pm 0.06}$ (27)	$19.31_{\pm 0.04}$ (26)	$929.0_{\pm 0.9}$ (25)	$918.1_{\pm 1.0}$ (25)	$3.5_{\pm 0.2}$ (29)	$3.7_{\pm 0.2}$ (29)	26.25 (28)
	GANomaly	$66.02_{\pm 0.58}$ (23)	$18.65_{\pm 0.25}$ (24)	$31.59_{\pm 0.38}$ (19)	$18.38_{\pm 0.16}$ (21)	$930.7_{\pm 6.0}$ (24)	$937.6_{\pm 3.7}$ (21)	$8.6_{\pm 0.8}$ (16)	$9.0_{\pm 0.7}$ (16)	20.50 (20)
	DROCC	$61.17_{\pm 0.63}$ (28)	$19.53_{\pm 0.20}$ (26)	$29.19_{\pm 0.33}$ (25)	$19.05_{\pm 0.23}$ (24)	$909.1_{\pm 5.3}$ (26)	$921.3_{\pm 6.3}$ (24)	$5.6_{\pm 0.4}$ (25)	$5.6_{\pm 0.2}$ (24)	25.25 (27)
	GOAD	$60.34_{\pm 0.46}$ (31)	$19.99_{\pm 0.23}$ (27)	$30.94_{\pm 0.31}$ (23)	$19.15_{\pm 0.27}$ (25)	$896.0_{\pm 5.9}$ (27)	$917.6_{\pm 6.6}$ (26)	$7.5_{\pm 0.5}$ (20)	$6.4_{\pm 0.4}$ (21)	25.00 (25)
	DDPM	$64.81_{\pm 0.12}$ (25)	$20.48_{\pm 0.09}$ (29)	$30.98_{\pm 0.11}$ (22)	$20.03_{\pm 0.10}$ (29)	$883.9_{\pm 2.3}$ (29)	$896.4_{\pm 2.4}$ (30)	$3.5_{\pm 0.5}$ (29)	$3.4_{\pm 0.3}$ (30)	27.88 (29)
	LODA	$61.85_{\pm 0.51}$ (27)	$20.42_{\pm 0.20}$ (28)	$27.40_{\pm 0.37}$ (30)	$20.14_{\pm 0.24}$ (30)	$890.6_{\pm 5.2}$ (28)	$897.9_{\pm 6.2}$ (29)	$5.0_{\pm 1.0}$ (26)	$4.9_{\pm 1.0}$ (25)	27.88 (29)
	DeepSVDD	$60.80_{\pm 0.35}$ (29)	$20.86_{\pm 0.22}$ (30)	$32.10_{\pm 0.19}$ (18)	$19.33_{\pm 0.28}$ (27)	$872.1_{\pm 5.7}$ (30)	$911.9_{\pm 7.0}$ (28)	$8.4_{\pm 0.5}$ (18)	$9.8_{\pm 0.4}$ (14)	24.25 (24)
	DAGMM	$60.50_{\pm 1.19}$ (30)	$22.54_{\pm 0.47}$ (31)	$25.24_{\pm 1.28}$ (31)	$22.75_{\pm 0.40}$ (31)	$831.8_{\pm 13.5}$ (31)	$826.1_{\pm 11.3}$ (31)	$3.0_{\pm 0.3}$ (31)	$3.0_{\pm 0.5}$ (31)	30.88 (31)
TFM-based	TACTIC-Clean	$75.24_{\pm 0.01}$ (5)	$12.22_{\pm 0.05}$ (5)	$41.70_{\pm 0.02}$ (7)	$11.66_{\pm 0.05}$ (4)	$1086.1_{\pm 1.4}$ (6)	$1100.2_{\pm 1.3}$ (4)	$17.9_{\pm 0.4}$ (7)	$19.0_{\pm 0.5}$ (6)	5.50 (4)
	TACTIC-Cont	$73.42_{\pm 0.01}$ (10)	$13.74_{\pm 0.03}$ (11)	$36.36_{\pm 0.02}$ (14)	$14.59_{\pm 0.04}$ (14)	$1048.0_{\pm 0.7}$ (11)	$1028.0_{\pm 1.0}$ (14)	$9.9_{\pm 0.3}$ (15)	$8.1_{\pm 0.2}$ (17)	13.25 (13)
	OUTFORMER	$74.16_{\pm 0.08}$ (7)	$13.32_{\pm 0.05}$ (10)	$43.00_{\pm 0.13}$ (3)	$12.19_{\pm 0.02}$ (6)	$1059.1_{\pm 1.4}$ (10)	$1087.3_{\pm 0.5}$ (5)	$19.9_{\pm 0.4}$ (4)	$21.5_{\pm 0.3}$ (3)	6.00 (5)
	FoMo-0D	$72.65_{\pm 0.04}$ (12)	$14.03_{\pm 0.09}$ (14)	$42.70_{\pm 0.13}$ (4)	$12.44_{\pm 0.10}$ (7)	$1041.1_{\pm 2.1}$ (13)	$1080.7_{\pm 2.4}$ (7)	$19.1_{\pm 0.5}$ (5)	$23.4_{\pm 0.4}$ (2)	8.00 (9)
	TabPFN-Extension	$73.62_{\pm 0.11}$ (8)	$12.29_{\pm 0.07}$ (6)	$37.09_{\pm 0.07}$ (13)	$13.43_{\pm 0.07}$ (11)	$1086.4_{\pm 1.6}$ (5)	$1058.8_{\pm 1.3}$ (11)	$24.8_{\pm 0.3}$ (3)	$19.1_{\pm 0.3}$ (5)	7.75 (7)
	TaskBridge	**$80.24_{\pm 0.07}$ (1)**	**$8.39_{\pm 0.10}$ (1)**	**$50.43_{\pm 0.10}$ (1)**	**$7.82_{\pm 0.06}$ (1)**	**$1189.5_{\pm 3.2}$ (1)**	**$1208.1_{\pm 2.2}$ (1)**	**$44.9_{\pm 1.3}$ (1)**	**$46.3_{\pm 1.6}$ (1)**	**1.00 (1)**

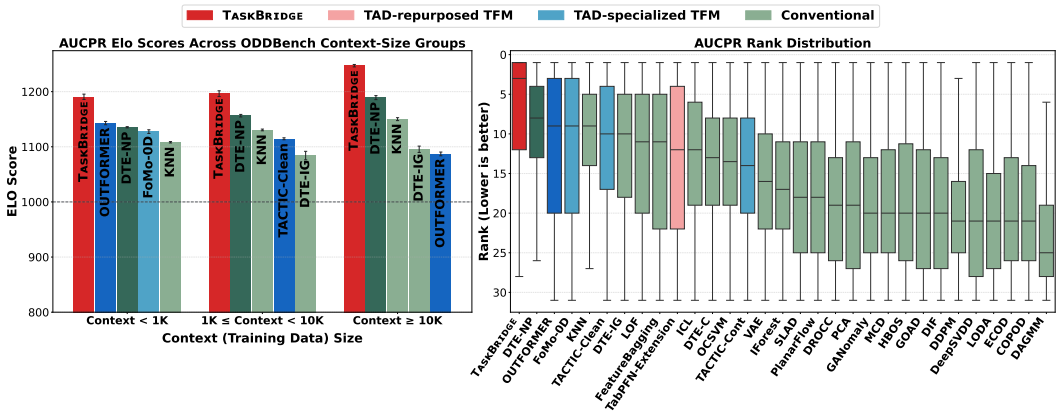


Figure 6: AUCPR performance comparison on ODDBench. (Extended results corresponding to Figure 4.) (Left) Elo scores across three dataset groups ($<$ 1K, 1K–10K, and $\geq$ 10K), showing the five methods with the highest average Elo scores in each group. (Right) Per-dataset rank distributions across all baselines; colors follow Figure 1. TaskBridge also achieves the highest Elo score across all dataset groups with different context size and the strongest overall rank distribution for AUCPR.

As a complementary check, Table 5 reports AUCROC and AUCPR Elo scores on the subset of approximately 720 ODDBench datasets for which all 31 methods successfully produce results, without requiring any imputation. TaskBridge still achieves the strongest performance on both metrics.

Figure 6 provides the corresponding AUCPR analysis to Figure 4 in the main paper, reporting Elo scores across ODDBench dataset groups stratified by context (training data) size and per-dataset rank distributions. We further provide the full Elo score comparisons for all datasets and each context-size group in Figures 7– 10.

Table 5: Elo scores on the subset of ODDBench datasets for which all 31 methods successfully produce results. The five methods with the highest AUCROC Elo scores are shown.

Method	AUCROC	AUCPR
TaskBridge	**$1188_{\pm 2.6}$**	**$1207_{\pm 2.8}$**
DTE-NP	$1167_{\pm 1.1}$	$1157_{\pm 1.0}$
KNN	$1142_{\pm 0.9}$	$1127_{\pm 0.9}$
TACTIC-Clean	$1103_{\pm 1.2}$	$1106_{\pm 1.4}$
FeatureBagging	$1084_{\pm 5.1}$	$1057_{\pm 5.6}$

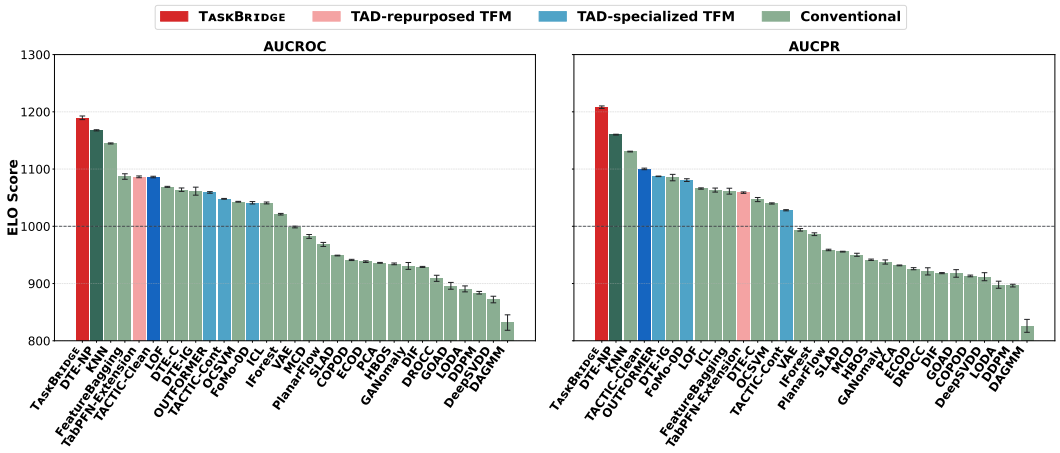


Figure 7: Elo score comparison across all baselines on all 790 ODDBench datasets. TASKBRIDGE achieves the highest Elo scores for both AUCROC and AUCPR, demonstrating consistently strong performance across diverse tabular datasets.

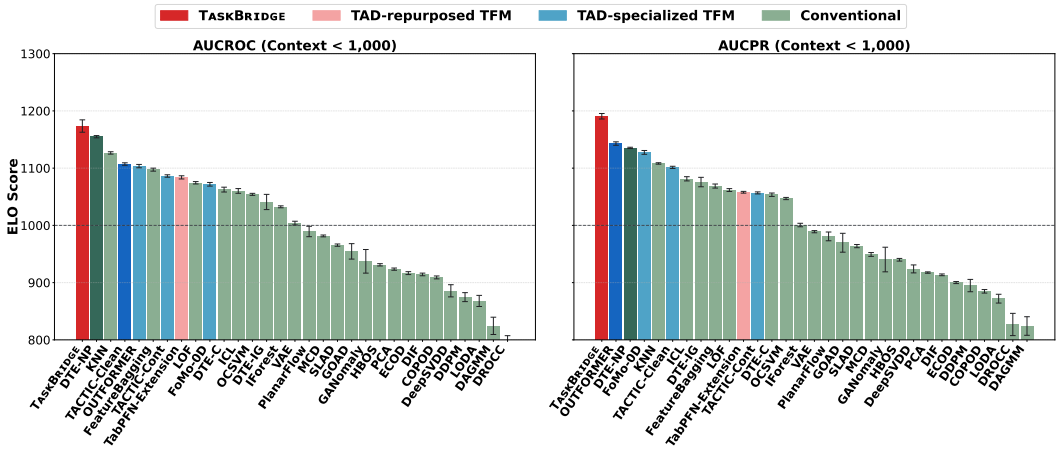


Figure 8: Elo score comparison across all baselines on partial ODDBench datasets with context (training data) size below 1K. TASKBRIDGE achieves the highest Elo scores for both AUCROC and AUCPR, demonstrating strong performance even with limited normality-anchoring evidence.

C.2 DETAILS OF ABLATION STUDIES

For all ablation studies, we vary only the component under investigation while keeping the remaining configuration of TASKBRIDGE fixed, using TabICLv2 as the pretrained TFM backbone.

For the task-family ablation, shown in Figure 5, Tables 6, 7, 8, and 9 report the detailed AUCROC and AUCPR results obtained on 790 ODDBench datasets by removing one to four task templates, respectively. Performance generally degrades as more tasks are removed, while the remaining families still retain competitive performance. These results support the complementary contribution of the instantiated virtual-task families and the robustness of the subsequent task-selection procedure.

For the task-selection ablation, *wo-Intra-Selection* removes candidate configuration selection within each task and instead uses the same default task configuration for all tasks. For the held-out split ablation, we change only the number of held-out splits from the default $S = 3$ to $S = 1$, while keeping all other settings unchanged.

For *wo-Inter-Selection*, we replace the systematic inter-task selection stage with random selection of a single task using its best intra-task configuration. Since the retention policy selected by the configuration search in Appendix B.1 is ALL, which retains all tasks passing the inter-task eligibility criterion, simply retaining all eligible tasks would still preserve part of the inter-task selection

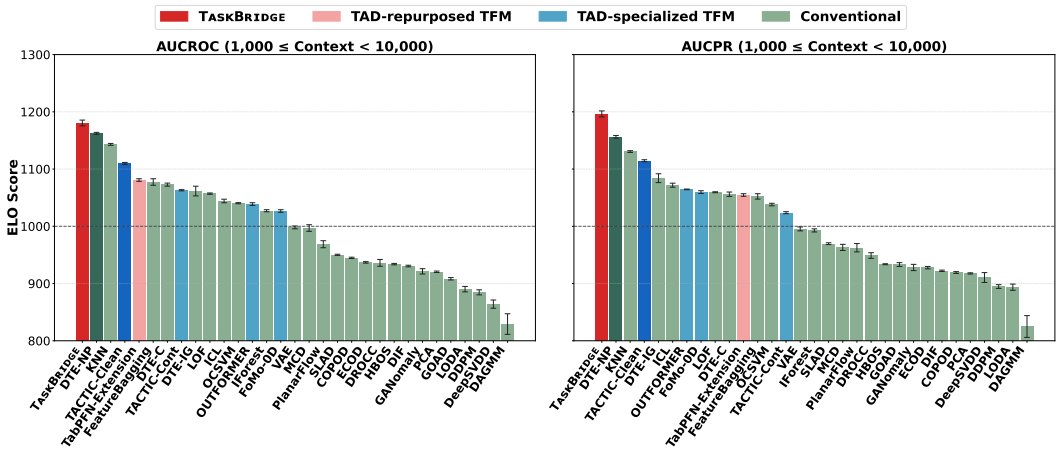


Figure 9: Elo score comparison across all baselines on partial ODDBench datasets with context (training data) size between 1K and 10K. TASKBRIDGE achieves the highest Elo scores for both AUCROC and AUCPR.

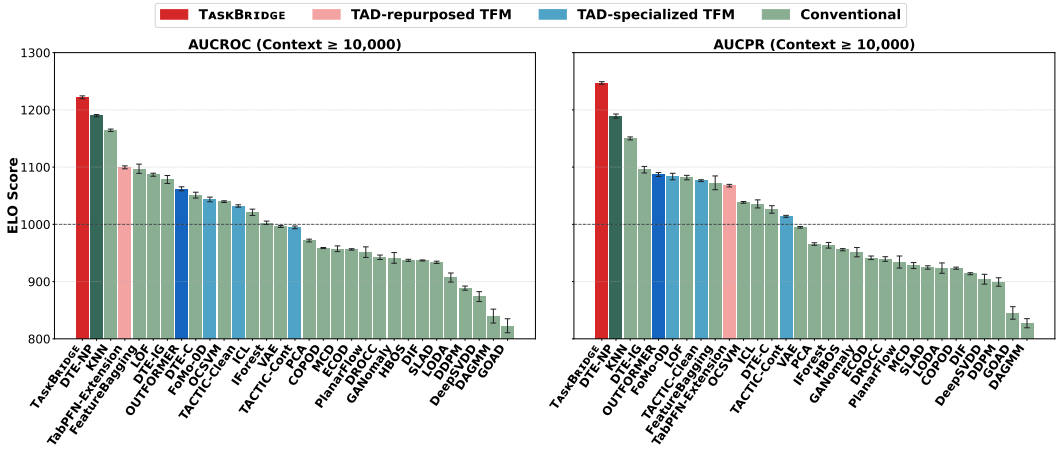


Figure 10: Elo score comparison across all baselines on partial ODDBench datasets with context (training data) size of at least 10K. TASKBRIDGE achieves the highest Elo scores for both AUCROC and AUCPR, maintaining strong performance with large amounts of normality-anchoring evidence.

mechanism. As multiple eligible tasks are available for most datasets, the random single-task variant provides a reference for assessing the contribution of systematic inter-task retention.

To further assess this contribution, we randomly retain increasing numbers of tasks and ensemble their scores. As shown in Table 10, performance generally improves as more complementary tasks are included, yet the full TASKBRIDGE with inter-task selection still achieves the strongest performance. Together with the task-family ablation, these results suggest that task complementarity and task-score aggregation can partially compensate for the absence of inter-task selection, while retaining tasks according to anomaly-detection-oriented predictive properties provides additional gains.

Overall, the full selection procedure achieves the strongest performance, while the $S = 1$ variant retains competitive detection performance with a substantially more efficient task-selection process.

C.3 INFERENCE EFFICIENCY

The main inference overhead of TASKBRIDGE arises from the hierarchical task-selection procedure. Once the tasks are selected, inference with the pretrained general-purpose TFM requires only milliseconds per sample, as shown in the rightmost panel of Figure 11. We evaluate the task-selection runtime using both our default configuration with three held-out splits and an efficient variant us-

Table 6: Ablation results when removing a single virtual task from the set of task templates. Single denotes single-attribute, Localized/Global denote localized/global (subspace) projection, Prototype denotes prototype-based organization, and Distributional denotes distributional extremity.

Remove-1	**Single**	**Localized**	**Global**	**Prototype**	**Distributional**
Avg. AUCROC	80.08	79.58	79.98	79.94	80.07
Avg. AUCPR	49.76	49.26	50.19	49.77	49.99

Table 7: Ablation results when removing two virtual tasks from the set of task templates. S denotes single-attribute, L/G denote localized/global (subspace) projection, P denotes prototype-based organization, and D denotes distributional extremity.

Remove-2	**S+L**	**S+G**	**S+P**	**S+D**	**L+G**	**L+P**	**L+D**	**G+P**	**G+D**	**P+D**
Avg. AUCROC	79.35	80.11	79.65	79.72	79.30	78.95	79.13	79.62	79.72	79.60
Avg. AUCPR	47.19	49.16	48.42	48.86	48.57	47.82	48.16	49.17	49.63	48.88

Table 8: Ablation results when removing three virtual tasks from the set of task templates. S denotes single-attribute, L/G denote localized/global (subspace) projection, P denotes prototype-based organization, and D denotes distributional extremity.

Remove-3	**S+L+G**	**S+L+P**	**S+L+D**	**S+G+P**	**S+G+D**	**S+P+D**	**L+G+P**	**L+G+D**	**L+P+D**	**G+P+D**
Avg. AUCROC	78.21	77.61	78.02	79.01	79.58	78.52	78.36	78.64	78.07	79.02
Avg. AUCPR	45.20	42.60	43.85	46.30	47.58	45.62	45.57	47.15	45.18	47.96

Table 9: Ablation results when removing four virtual tasks from the set of task templates. For brevity, each variant is labeled by the remaining task. Single denotes single-attribute, Localized/Global denote localized/global (subspace) projection, Prototype denotes prototype-based organization, and Distributional denotes distributional extremity.

Remove-4	**Single**	**Localized**	**Global**	**Prototype**	**Distributional**
Avg. AUCROC	77.12	77.99	76.11	76.15	76.79
Avg. AUCPR	42.33	42.46	36.51	39.01	38.76

Table 10: Further ablation of inter-task selection by randomly retaining increasing numbers of tasks after intra-task selection for score ensembling. Across all ODDBench datasets, performance improves as more tasks are retained, while the full TASKBRIDGE with inter-task selection achieves the best performance.

	2 Remaining Tasks	**3 Remaining Tasks**	**4 Remaining Tasks**	**Full TASKBRIDGE**
Avg. AUCROC	78.11	79.38	79.89	**80.16**
Avg. AUCPR	44.95	48.43	49.66	**50.43**

ing a single split, which also achieves the strongest overall performance against 31 baselines on ODDBench.

As shown in the leftmost panels of Figure 11, the task-selection time of the default TASKBRIDGE remains below one minute for most datasets, even when the context size approaches the 10^4 scale. Reducing the number of held-out splits to $S = 1$ further lowers this cost, requiring less than 20 seconds for most datasets across diverse context sizes and dimensionalities. Importantly, task selection is performed only once per dataset and requires neither dataset-specific hyperparameter search nor parameter optimization, thereby preserving efficient TAD inference through in-context learning.

Overall, this one-time selection cost preserves TASKBRIDGE as an efficient TAD-repurposing framework. As shown in Figure 1, TASKBRIDGE remains substantially more efficient than the existing TabPFN-Extension even when its task-selection cost is included in the comparison.

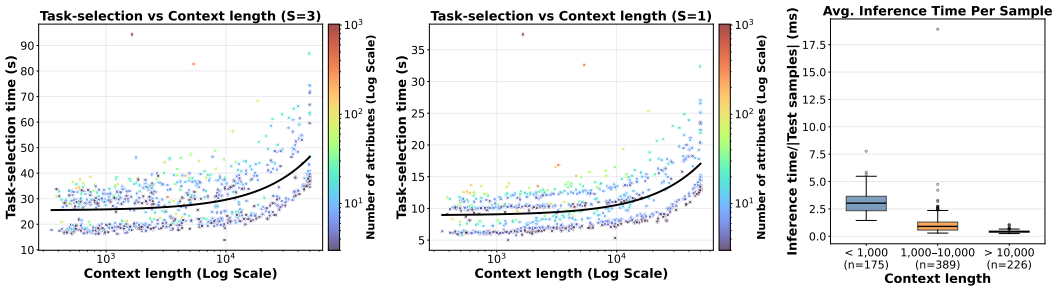


Figure 11: Inference-efficiency analysis of TASKBRIDGE. (Left, middle) Task-selection time across ODDBench datasets using 3 (default) and 1 held-out splits, respectively, as a function of context length and the number of attributes. (Right) Average inference time per test sample after task selection, grouped by context length; TASKBRIDGE maintains millisecond-level inference across all groups. Here, n denotes the number of datasets in each group: $<$ 1,000, 1,000–10,000, and $>$ 10,000 context length. Overall, TASKBRIDGE preserves efficient inference with TAD-repurposed TFMs.

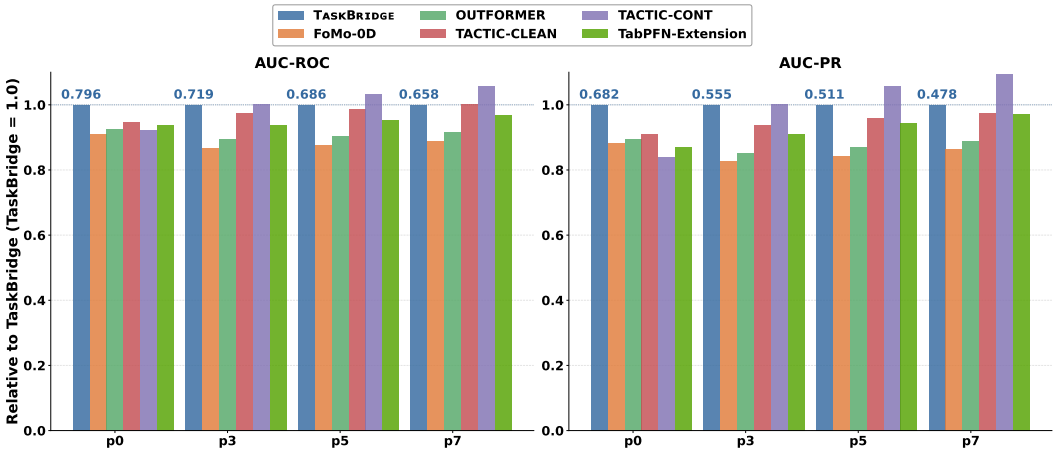


Figure 12: Robustness to context contamination on ODDBench. AUCROC and AUCPR performance are shown relative to TASKBRIDGE at each contamination level (TASKBRIDGE $= 1$), with its absolute performance reported above each group. pk denotes the setting in which $k\%$ of the context samples are replaced with anomalous samples. TASKBRIDGE remains competitive with TFM-based TAD baselines as contamination increases, while the contamination-specific TACTIC-CONT exhibits the strongest robustness under heavier contamination.

C.4 ROBUSTNESS ANALYSIS ON CONTEXT CONTAMINATION

Considering practical scenarios in which the training/context data may be contaminated, we evaluate the robustness of TASKBRIDGE to context contamination, which can interfere with normality anchoring. We construct a fixed subset of 295 ODDBench datasets that support the most severe 10% contamination setting without removing more than 50% of the test anomalies. For each dataset, we replace a fraction of normality-aware context samples with randomly selected anomalies drawn from the test/query split. We evaluate contamination rates of 0%, 3%, 5%, and 7%. To ensure a fair comparison across contamination levels, all settings use the same test/query set after excluding the fixed anomaly pool used for contamination. Normal query samples are then downsampled to keep the anomaly ratio in the test split consistent across contamination levels. We compare TASKBRIDGE under each contamination level with TFM-based TAD baselines, including TACTIC-CONT, which is specifically pretrained to handle contaminated contexts.

As shown in Figure 12, the performance of all methods generally degrades as context contamination increases. Nevertheless, TASKBRIDGE remains competitive and generally outperforms the TFM-based baselines that are not specifically trained for contaminated contexts. TACTIC-CONT shows

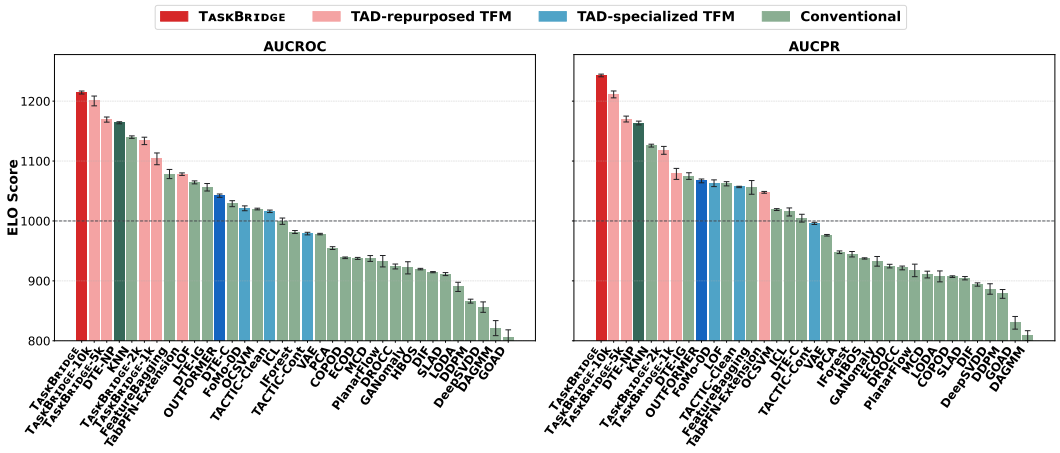


Figure 13: Elo score comparison across all baselines on 226 ODDBench datasets, including four TASKBRIDGE variants with restricted context budgets. Each TASKBRIDGE-iK variant limits the context size to iK samples. All variants, including the 1K setting, retain strong performance relative to the other baselines.

the strongest robustness under heavier contamination, consistent with its contamination-aware pretraining. These results suggest that context contamination can impair the normality anchoring of virtual supervised tasks, while the data-adaptive task-selection procedure still retains relatively suitable tasks under corrupted contexts. Improving the construction and selection of virtual tasks under contaminated contexts therefore constitutes an important direction for future work.

C.5 ROBUSTNESS ANALYSIS ON CONTEXT SIZE

To evaluate the robustness of TASKBRIDGE to the amount of context evidence available for normality-anchored virtual task construction, data-adaptive task selection, and in-context prediction, we conduct a controlled context-size analysis. We select the 226 ODDBench datasets whose original normality-aware context contains more than 10K samples. For each dataset, we randomly subsample the context to fixed budgets of 10K, 5K, 2K, and 1K samples. Each subsampled context is used throughout the entire TASKBRIDGE pipeline, including normality-anchored task construction, data-adaptive task selection, and in-context prediction with the pretrained general-purpose TFM. We compare these variants with the original full-context TASKBRIDGE and the 30 baselines using Elo scores computed over the same 226 datasets.

As shown in Figure 13, performance gradually decreases as the available context budget is reduced, indicating that richer context evidence benefits the entire TASKBRIDGE pipeline. Nevertheless, TASKBRIDGE remains highly competitive even under substantial context reduction. In particular, the 10K and 5K variants outperform all external baselines in both AUCROC and AUCPR Elo scores, while the 2K and 1K variants remain among the top-performing methods and ahead of the TFM-based TAD baselines. These results complement the context-size analysis in Section 4.1, where TASKBRIDGE also exhibits strong performance across datasets with naturally varying context sizes. This robustness suggests that complementary normality-anchored task constructions, which probe different predictive structures in the context, together with data-adaptive task selection, can preserve informative task structures even under limited context budgets, while benefiting from the context-efficient predictive capability of pretrained general-purpose TFMs. Together with the inference analysis in Appendix C.3, these results further indicate that TASKBRIDGE can operate efficiently with small context budgets while maintaining robust anomaly detection performance.

C.6 GENERALIZATION TO DIFFERENT PRETRAINED PFN-BASED TFM BACKBONES

To assess whether TASKBRIDGE generalizes beyond the TabICLv2 backbone used in our main experiments (Qu et al., 2026), we replace it with two alternative pretrained general-purpose TFMs: TabPFN v2.6[4] and TabPFN v3 (Grinsztajn et al., 2026b), both of which also exhibit strong pre-

[4] https://huggingface.co/Prior-Labs/tabpfn_2_6

Table 11: Backbone generalization of TASKBRIDGE with TabPFN v2.6. Total Rank is computed as in Table 4. With this alternative pretrained general-purpose TFM backbone, TASKBRIDGE remains highly competitive among 31 methods.

	Methods	Avg. AUCROC ↑	Avg. Rank (AUCROC) ↓	Avg. AUCPR ↑	Avg. Rank (AUCPR) ↓	ELO (AUCROC) ↑	ELO (AUCPR) ↑	Top-3 Ratio (AUCROC) ↑	Top-3 Ratio (AUCPR) ↑	Total Rank ↓
Conventional	DTE-NP	**76.96**$_{\pm 0.01}$ **(1)**	**8.90**$_{\pm 0.02}$ **(1)**	**43.63**$_{\pm 0.01}$ **(1)**	**9.18**$_{\pm 0.03}$ **(1)**	**1172.2**$_{\pm 0.9}$ **(1)**	**1164.6**$_{\pm 0.9}$ **(1)**	**27.9**$_{\pm 0.5}$ **(1)**	22.2$_{\pm 0.5}$ (4)	**1.38 (1)**
	KNN	76.25$_{\pm 0.01}$ (2)	9.65$_{\pm 0.02}$ (2)	42.41$_{\pm 0.01}$ (4)	10.20$_{\pm 0.03}$ (2)	1149.0$_{\pm 0.9}$ (2)	1134.8$_{\pm 1.0}$ (2)	18.6$_{\pm 0.6}$ (8)	14.9$_{\pm 0.5}$ (10)	4.00 (2)
	FeatureBagging	75.59$_{\pm 0.16}$ (3)	11.97$_{\pm 0.21}$ (3)	39.30$_{\pm 0.56}$ (10)	13.00$_{\pm 0.22}$ (10)	1091.1$_{\pm 5.0}$ (3)	1065.9$_{\pm 5.3}$ (10)	19.8$_{\pm 0.9}$ (6)	18.9$_{\pm 1.3}$ (8)	6.62 (6)
	LOF	74.17$_{\pm 0.02}$ (6)	12.64$_{\pm 0.04}$ (6)	39.97$_{\pm 0.02}$ (9)	12.77$_{\pm 0.05}$ (7)	1073.6$_{\pm 1.0}$ (6)	1070.9$_{\pm 1.2}$ (8)	14.4$_{\pm 0.6}$ (10)	16.3$_{\pm 1.0}$ (9)	7.62 (9)
	DTE-C	73.46$_{\pm 0.05}$ (9)	12.85$_{\pm 0.12}$ (7)	38.16$_{\pm 0.09}$ (11)	13.58$_{\pm 0.14}$ (12)	1068.1$_{\pm 3.2}$ (7)	1051.0$_{\pm 3.5}$ (12)	11.3$_{\pm 0.7}$ (12)	9.7$_{\pm 0.6}$ (15)	10.62 (11)
	DTE-IG	73.34$_{\pm 0.43}$ (11)	12.91$_{\pm 0.30}$ (8)	41.92$_{\pm 0.34}$ (6)	11.96$_{\pm 0.23}$ (4)	1065.8$_{\pm 7.0}$ (9)	1089.8$_{\pm 5.5}$ (5)	16.3$_{\pm 1.5}$ (9)	20.0$_{\pm 1.1}$ (5)	7.12 (7)
	OCSVM	71.88$_{\pm 0.01}$ (13)	13.73$_{\pm 0.03}$ (12)	37.18$_{\pm 0.01}$ (12)	13.86$_{\pm 0.04}$ (13)	1046.6$_{\pm 0.8}$ (12)	1044.2$_{\pm 0.9}$ (13)	8.8$_{\pm 0.6}$ (17)	7.4$_{\pm 0.3}$ (19)	13.88 (14)
	ICL	71.82$_{\pm 0.10}$ (14)	13.86$_{\pm 0.07}$ (14)	40.07$_{\pm 0.15}$ (8)	12.92$_{\pm 0.15}$ (9)	1044.8$_{\pm 1.4}$ (14)	1067.8$_{\pm 3.5}$ (9)	11.9$_{\pm 0.2}$ (11)	13.2$_{\pm 0.4}$ (11)	11.25 (12)
	IForest	71.26$_{\pm 0.05}$ (15)	14.92$_{\pm 0.06}$ (15)	31.26$_{\pm 0.10}$ (20)	16.37$_{\pm 0.10}$ (16)	1024.9$_{\pm 1.4}$ (15)	990.7$_{\pm 2.4}$ (16)	5.1$_{\pm 0.6}$ (27)	4.5$_{\pm 0.1}$ (27)	18.88 (18)
	MCD	69.10$_{\pm 0.19}$ (17)	16.41$_{\pm 0.15}$ (17)	31.10$_{\pm 0.22}$ (21)	17.75$_{\pm 0.11}$ (19)	986.2$_{\pm 3.3}$ (17)	954.5$_{\pm 2.6}$ (19)	11.3$_{\pm 0.7}$ (13)	10.3$_{\pm 0.6}$ (12)	16.88 (17)
	VAE	70.20$_{\pm 0.04}$ (16)	15.60$_{\pm 0.07}$ (16)	35.34$_{\pm 0.07}$ (15)	15.80$_{\pm 0.10}$ (15)	1002.9$_{\pm 1.6}$ (16)	998.8$_{\pm 2.2}$ (15)	7.9$_{\pm 0.4}$ (19)	6.7$_{\pm 0.3}$ (21)	16.62 (16)
	PlanarFlow	68.11$_{\pm 0.40}$ (18)	16.99$_{\pm 0.13}$ (18)	32.95$_{\pm 0.17}$ (17)	17.40$_{\pm 0.05}$ (17)	971.8$_{\pm 3.5}$ (18)	962.6$_{\pm 1.5}$ (17)	11.2$_{\pm 0.6}$ (14)	10.2$_{\pm 0.8}$ (13)	16.50 (15)
	SLAD	66.87$_{\pm 0.05}$ (21)	17.71$_{\pm 0.03}$ (19)	33.49$_{\pm 0.09}$ (16)	17.44$_{\pm 0.04}$ (18)	953.0$_{\pm 0.6}$ (19)	960.0$_{\pm 0.8}$ (18)	6.9$_{\pm 0.1}$ (23)	7.3$_{\pm 0.2}$ (20)	19.25 (19)
	COPOD	67.20$_{\pm 0.01}$ (20)	18.22$_{\pm 0.05}$ (20)	27.99$_{\pm 0.01}$ (29)	19.35$_{\pm 0.06}$ (28)	945.7$_{\pm 1.3}$ (20)	918.3$_{\pm 1.5}$ (27)	3.8$_{\pm 0.3}$ (28)	4.3$_{\pm 0.2}$ (28)	25.00 (25)
	PCA	64.50$_{\pm 0.03}$ (26)	18.37$_{\pm 0.02}$ (22)	29.14$_{\pm 0.02}$ (26)	18.53$_{\pm 0.04}$ (22)	939.9$_{\pm 0.6}$ (22)	936.4$_{\pm 0.8}$ (22)	6.3$_{\pm 0.3}$ (24)	5.0$_{\pm 0.3}$ (26)	23.75 (23)
	ECOD	68.10$_{\pm 0.01}$ (19)	18.37$_{\pm 0.06}$ (21)	28.25$_{\pm 0.02}$ (28)	18.85$_{\pm 0.08}$ (23)	942.6$_{\pm 1.5}$ (21)	930.9$_{\pm 1.9}$ (23)	6.9$_{\pm 0.4}$ (22)	6.2$_{\pm 0.4}$ (23)	22.50 (22)
	HBOS	66.58$_{\pm 0.02}$ (22)	18.48$_{\pm 0.06}$ (23)	29.90$_{\pm 0.03}$ (24)	18.19$_{\pm 0.05}$ (20)	938.5$_{\pm 1.6}$ (23)	946.1$_{\pm 1.1}$ (20)	7.0$_{\pm 0.3}$ (21)	7.6$_{\pm 0.5}$ (18)	21.38 (21)
	DIF	65.16$_{\pm 0.02}$ (24)	18.74$_{\pm 0.04}$ (25)	28.53$_{\pm 0.06}$ (27)	19.19$_{\pm 0.04}$ (26)	933.1$_{\pm 1.0}$ (25)	922.4$_{\pm 1.0}$ (25)	3.6$_{\pm 0.1}$ (29)	3.8$_{\pm 0.2}$ (29)	26.25 (28)
	GANomaly	66.02$_{\pm 0.58}$ (23)	18.52$_{\pm 0.26}$ (24)	31.59$_{\pm 0.38}$ (19)	18.24$_{\pm 0.16}$ (21)	934.7$_{\pm 6.0}$ (24)	942.3$_{\pm 3.7}$ (21)	9.0$_{\pm 0.9}$ (16)	9.6$_{\pm 0.7}$ (16)	20.50 (20)
	DROCC	61.17$_{\pm 0.63}$ (28)	19.41$_{\pm 0.20}$ (26)	29.19$_{\pm 0.33}$ (25)	18.91$_{\pm 0.23}$ (24)	913.0$_{\pm 5.2}$ (26)	925.9$_{\pm 6.2}$ (24)	5.9$_{\pm 0.6}$ (25)	6.1$_{\pm 0.4}$ (24)	25.25 (27)
	GOAD	60.34$_{\pm 0.46}$ (31)	19.91$_{\pm 0.23}$ (27)	30.94$_{\pm 0.31}$ (23)	19.03$_{\pm 0.27}$ (25)	899.2$_{\pm 5.9}$ (27)	921.8$_{\pm 6.6}$ (26)	7.7$_{\pm 0.5}$ (20)	6.5$_{\pm 0.4}$ (22)	25.12 (26)
	DDPM	64.81$_{\pm 0.12}$ (25)	20.37$_{\pm 0.08}$ (29)	30.98$_{\pm 0.11}$ (22)	19.90$_{\pm 0.10}$ (29)	887.9$_{\pm 2.0}$ (29)	900.9$_{\pm 2.5}$ (30)	3.5$_{\pm 0.5}$ (30)	3.4$_{\pm 0.3}$ (30)	28.00 (30)
	LODA	61.85$_{\pm 0.51}$ (27)	20.30$_{\pm 0.20}$ (28)	27.40$_{\pm 0.37}$ (30)	20.00$_{\pm 0.23}$ (30)	894.8$_{\pm 5.1}$ (28)	902.9$_{\pm 6.0}$ (29)	5.1$_{\pm 1.0}$ (26)	5.1$_{\pm 0.9}$ (25)	27.88 (29)
	DeepSVDD	60.80$_{\pm 0.35}$ (29)	20.78$_{\pm 0.23}$ (30)	32.10$_{\pm 0.19}$ (18)	19.22$_{\pm 0.28}$ (27)	875.3$_{\pm 5.9}$ (30)	915.8$_{\pm 7.0}$ (28)	8.4$_{\pm 0.5}$ (18)	10.0$_{\pm 0.7}$ (14)	24.25 (24)
	DAGMM	60.50$_{\pm 1.19}$ (30)	22.44$_{\pm 0.48}$ (31)	25.24$_{\pm 1.28}$ (31)	22.64$_{\pm 0.41}$ (31)	836.1$_{\pm 13.7}$ (31)	831.0$_{\pm 11.5}$ (31)	3.1$_{\pm 0.4}$ (31)	3.1$_{\pm 0.7}$ (31)	30.88 (31)
TFM-based	TACTIC-Clean	75.22$_{\pm 0.01}$ (5)	12.03$_{\pm 0.05}$ (4)	41.67$_{\pm 0.02}$ (7)	11.47$_{\pm 0.04}$ (3)	1090.4$_{\pm 1.5}$ (4)	1104.8$_{\pm 1.3}$ (3)	18.7$_{\pm 0.3}$ (7)	19.8$_{\pm 0.3}$ (6)	4.88 (3)
	TACTIC-Cont	73.40$_{\pm 0.01}$ (10)	13.59$_{\pm 0.02}$ (11)	36.33$_{\pm 0.02}$ (14)	14.43$_{\pm 0.04}$ (14)	1051.8$_{\pm 0.6}$ (11)	1032.1$_{\pm 1.0}$ (14)	10.4$_{\pm 0.2}$ (15)	8.3$_{\pm 0.4}$ (17)	13.25 (13)
	OUTFORMER	74.16$_{\pm 0.08}$ (7)	13.13$_{\pm 0.05}$ (10)	43.00$_{\pm 0.13}$ (2)	11.99$_{\pm 0.03}$ (5)	1063.6$_{\pm 1.3}$ (10)	1091.9$_{\pm 0.6}$ (4)	20.8$_{\pm 0.6}$ (4)	23.0$_{\pm 0.4}$ (3)	5.62 (5)
	FoMo-0D	72.65$_{\pm 0.04}$ (12)	13.85$_{\pm 0.09}$ (13)	42.70$_{\pm 0.13}$ (3)	12.24$_{\pm 0.11}$ (6)	1045.5$_{\pm 2.1}$ (13)	1085.5$_{\pm 2.5}$ (6)	20.6$_{\pm 0.4}$ (5)	24.9$_{\pm 0.7}$ (2)	7.50 (8)
	TabPFN-Extension	73.61$_{\pm 0.11}$ (8)	12.12$_{\pm 0.07}$ (5)	37.08$_{\pm 0.07}$ (13)	13.25$_{\pm 0.07}$ (11)	1090.3$_{\pm 1.6}$ (5)	1063.0$_{\pm 1.4}$ (11)	25.5$_{\pm 0.1}$ (3)	19.8$_{\pm 0.3}$ (6)	7.75 (10)
	TASKBRIDGE	75.23$_{\pm 0.06}$ (4)	13.03$_{\pm 0.05}$ (9)	42.27$_{\pm 0.14}$ (5)	12.84$_{\pm 0.08}$ (8)	1067.8$_{\pm 1.1}$ (8)	1072.5$_{\pm 2.0}$ (7)	26.2$_{\pm 0.4}$ (2)	**27.7**$_{\pm 0.5}$ **(1)**	5.50 (4)

Table 12: Backbone generalization of TASKBRIDGE with TabPFN v3. Total Rank is computed as in Table 4. TASKBRIDGE remains consistently strong with the alternative pretrained general-purpose TFM backbone, achieving the second-best overall Total Rank among 31 methods.

	Methods	Avg. AUCROC ↑	Avg. Rank (AUCROC) ↓	Avg. AUCPR ↑	Avg. Rank (AUCPR) ↓	ELO (AUCROC) ↑	ELO (AUCPR) ↑	Top-3 Ratio (AUCROC) ↑	Top-3 Ratio (AUCPR) ↑	Total Rank ↓
Conventional	DTE-NP	**76.96**$_{\pm 0.01}$ **(1)**	**8.93**$_{\pm 0.03}$ **(1)**	43.63$_{\pm 0.01}$ (2)	**9.19**$_{\pm 0.03}$ **(1)**	**1171.5**$_{\pm 1.0}$ **(1)**	**1164.2**$_{\pm 0.9}$ **(1)**	**27.4**$_{\pm 0.6}$ **(1)**	21.5$_{\pm 0.5}$ (4)	**1.50 (1)**
	KNN	76.25$_{\pm 0.01}$ (2)	9.67$_{\pm 0.03}$ (2)	42.41$_{\pm 0.01}$ (5)	10.21$_{\pm 0.03}$ (2)	1148.4$_{\pm 1.2}$ (2)	1134.5$_{\pm 0.8}$ (2)	18.7$_{\pm 0.6}$ (8)	14.8$_{\pm 0.7}$ (10)	4.12 (3)
	FeatureBagging	75.59$_{\pm 0.16}$ (4)	11.99$_{\pm 0.20}$ (3)	39.30$_{\pm 0.56}$ (10)	13.01$_{\pm 0.21}$ (10)	1090.7$_{\pm 4.7}$ (4)	1065.7$_{\pm 5.1}$ (10)	20.1$_{\pm 1.2}$ (6)	19.2$_{\pm 1.3}$ (8)	6.88 (6)
	LOF	74.17$_{\pm 0.02}$ (6)	12.68$_{\pm 0.04}$ (7)	39.97$_{\pm 0.02}$ (9)	12.79$_{\pm 0.05}$ (8)	1072.6$_{\pm 1.1}$ (7)	1070.5$_{\pm 1.1}$ (8)	14.4$_{\pm 0.7}$ (10)	15.9$_{\pm 0.9}$ (9)	8.00 (10)
	DTE-C	73.46$_{\pm 0.05}$ (9)	12.88$_{\pm 0.12}$ (8)	38.16$_{\pm 0.09}$ (11)	13.59$_{\pm 0.14}$ (12)	1067.3$_{\pm 3.2}$ (8)	1050.7$_{\pm 3.6}$ (12)	11.3$_{\pm 0.7}$ (13)	9.8$_{\pm 0.4}$ (15)	11.00 (11)
	DTE-IG	73.34$_{\pm 0.43}$ (11)	12.93$_{\pm 0.30}$ (9)	41.92$_{\pm 0.34}$ (6)	11.97$_{\pm 0.23}$ (5)	1065.2$_{\pm 7.0}$ (9)	1089.4$_{\pm 5.5}$ (6)	16.5$_{\pm 1.1}$ (9)	20.2$_{\pm 0.9}$ (6)	7.62 (7)
	OCSVM	71.88$_{\pm 0.01}$ (13)	13.76$_{\pm 0.02}$ (12)	37.18$_{\pm 0.01}$ (12)	13.90$_{\pm 0.04}$ (13)	1045.9$_{\pm 0.7}$ (12)	1043.3$_{\pm 0.8}$ (13)	8.5$_{\pm 0.4}$ (18)	7.3$_{\pm 0.3}$ (18)	13.88 (14)
	ICL	71.82$_{\pm 0.10}$ (14)	13.88$_{\pm 0.07}$ (14)	40.07$_{\pm 0.15}$ (8)	12.95$_{\pm 0.15}$ (9)	1044.2$_{\pm 1.4}$ (14)	1067.3$_{\pm 3.6}$ (9)	11.8$_{\pm 0.4}$ (11)	13.3$_{\pm 0.6}$ (11)	11.25 (12)
	IForest	71.26$_{\pm 0.05}$ (15)	14.95$_{\pm 0.06}$ (15)	31.26$_{\pm 0.10}$ (20)	16.40$_{\pm 0.10}$ (16)	1024.1$_{\pm 1.4}$ (15)	989.8$_{\pm 2.4}$ (16)	4.9$_{\pm 0.5}$ (27)	4.4$_{\pm 0.1}$ (27)	18.88 (18)
	MCD	69.10$_{\pm 0.19}$ (17)	16.43$_{\pm 0.15}$ (17)	31.10$_{\pm 0.22}$ (21)	17.77$_{\pm 0.12}$ (19)	985.7$_{\pm 3.3}$ (17)	953.9$_{\pm 2.7}$ (19)	11.4$_{\pm 0.8}$ (12)	10.5$_{\pm 0.5}$ (12)	16.75 (17)
	VAE	70.20$_{\pm 0.04}$ (16)	15.64$_{\pm 0.08}$ (16)	35.34$_{\pm 0.07}$ (15)	15.86$_{\pm 0.10}$ (15)	1001.9$_{\pm 1.8}$ (16)	997.5$_{\pm 2.3}$ (15)	7.8$_{\pm 0.4}$ (19)	6.7$_{\pm 0.4}$ (21)	16.62 (16)
	PlanarFlow	68.11$_{\pm 0.40}$ (18)	17.01$_{\pm 0.13}$ (18)	32.95$_{\pm 0.17}$ (17)	17.43$_{\pm 0.05}$ (17)	971.3$_{\pm 3.5}$ (18)	962.0$_{\pm 1.3}$ (17)	11.3$_{\pm 0.7}$ (14)	10.5$_{\pm 0.8}$ (12)	16.38 (15)
	SLAD	66.87$_{\pm 0.05}$ (21)	17.75$_{\pm 0.03}$ (19)	33.49$_{\pm 0.09}$ (16)	17.48$_{\pm 0.03}$ (18)	951.9$_{\pm 0.7}$ (19)	959.1$_{\pm 0.7}$ (18)	6.8$_{\pm 0.2}$ (22)	7.2$_{\pm 0.2}$ (19)	19.00 (19)
	COPOD	67.20$_{\pm 0.01}$ (20)	18.26$_{\pm 0.05}$ (20)	27.99$_{\pm 0.01}$ (29)	19.40$_{\pm 0.06}$ (28)	944.7$_{\pm 1.2}$ (20)	917.0$_{\pm 1.6}$ (27)	4.1$_{\pm 0.3}$ (28)	4.3$_{\pm 0.2}$ (28)	25.00 (25)
	PCA	64.50$_{\pm 0.03}$ (26)	18.40$_{\pm 0.02}$ (21)	29.14$_{\pm 0.02}$ (26)	18.57$_{\pm 0.04}$ (22)	939.1$_{\pm 0.6}$ (22)	935.3$_{\pm 0.8}$ (22)	6.3$_{\pm 0.1}$ (24)	4.9$_{\pm 0.3}$ (26)	23.62 (23)
	ECOD	68.10$_{\pm 0.01}$ (19)	18.40$_{\pm 0.06}$ (22)	28.25$_{\pm 0.02}$ (28)	18.89$_{\pm 0.08}$ (23)	941.7$_{\pm 1.4}$ (21)	929.9$_{\pm 2.0}$ (23)	6.9$_{\pm 0.4}$ (21)	6.7$_{\pm 0.5}$ (22)	22.38 (22)
	HBOS	66.58$_{\pm 0.02}$ (22)	18.53$_{\pm 0.06}$ (23)	29.90$_{\pm 0.03}$ (24)	18.24$_{\pm 0.05}$ (20)	937.2$_{\pm 1.5}$ (23)	944.8$_{\pm 1.2}$ (20)	6.8$_{\pm 0.3}$ (22)	7.1$_{\pm 0.4}$ (20)	21.75 (21)
	DIF	65.16$_{\pm 0.02}$ (24)	18.77$_{\pm 0.04}$ (25)	28.53$_{\pm 0.06}$ (27)	19.21$_{\pm 0.04}$ (26)	932.2$_{\pm 0.9}$ (25)	921.7$_{\pm 0.9}$ (25)	3.5$_{\pm 0.1}$ (29)	3.9$_{\pm 0.2}$ (29)	26.25 (28)
	GANomaly	66.02$_{\pm 0.58}$ (23)	18.56$_{\pm 0.25}$ (24)	31.59$_{\pm 0.38}$ (19)	18.27$_{\pm 0.16}$ (21)	933.8$_{\pm 5.8}$ (24)	941.3$_{\pm 3.7}$ (21)	9.0$_{\pm 1.0}$ (16)	9.6$_{\pm 0.7}$ (16)	20.50 (20)
	DROCC	61.17$_{\pm 0.63}$ (28)	19.43$_{\pm 0.19}$ (26)	29.19$_{\pm 0.33}$ (25)	18.94$_{\pm 0.23}$ (24)	912.5$_{\pm 5.2}$ (26)	925.0$_{\pm 6.2}$ (24)	5.8$_{\pm 0.5}$ (25)	6.1$_{\pm 0.4}$ (24)	25.25 (26)
	GOAD	60.34$_{\pm 0.46}$ (31)	19.93$_{\pm 0.24}$ (27)	30.94$_{\pm 0.31}$ (23)	19.07$_{\pm 0.27}$ (25)	898.4$_{\pm 6.1}$ (27)	920.8$_{\pm 6.7}$ (26)	7.7$_{\pm 0.6}$ (20)	6.5$_{\pm 0.4}$ (23)	25.25 (26)
	DDPM	64.81$_{\pm 0.12}$ (25)	20.39$_{\pm 0.09}$ (29)	30.98$_{\pm 0.11}$ (22)	19.93$_{\pm 0.09}$ (29)	887.2$_{\pm 2.1}$ (29)	900.2$_{\pm 2.2}$ (30)	3.5$_{\pm 0.5}$ (30)	3.3$_{\pm 0.3}$ (30)	28.00 (30)
	LODA	61.85$_{\pm 0.51}$ (27)	20.33$_{\pm 0.19}$ (28)	27.40$_{\pm 0.37}$ (30)	20.04$_{\pm 0.24}$ (30)	894.0$_{\pm 5.0}$ (28)	901.6$_{\pm 6.1}$ (29)	5.1$_{\pm 0.9}$ (26)	5.0$_{\pm 1.0}$ (25)	27.88 (29)
	DeepSVDD	60.80$_{\pm 0.35}$ (29)	20.81$_{\pm 0.22}$ (30)	32.10$_{\pm 0.19}$ (18)	19.25$_{\pm 0.28}$ (27)	874.4$_{\pm 5.8}$ (30)	915.0$_{\pm 6.9}$ (28)	8.5$_{\pm 0.5}$ (17)	10.2$_{\pm 0.4}$ (14)	24.12 (24)
	DAGMM	60.50$_{\pm 1.19}$ (30)	22.47$_{\pm 0.47}$ (31)	25.24$_{\pm 1.28}$ (31)	22.68$_{\pm 0.40}$ (31)	835.1$_{\pm 13.4}$ (31)	829.5$_{\pm 11.2}$ (31)	3.2$_{\pm 0.3}$ (31)	3.1$_{\pm 0.6}$ (31)	30.88 (31)
TFM-based	TACTIC-Clean	75.22$_{\pm 0.01}$ (5)	12.08$_{\pm 0.05}$ (5)	41.68$_{\pm 0.02}$ (7)	11.50$_{\pm 0.04}$ (3)	1089.4$_{\pm 1.4}$ (6)	1104.2$_{\pm 1.3}$ (3)	18.8$_{\pm 0.3}$ (7)	19.8$_{\pm 0.3}$ (7)	5.38 (4)
	TACTIC-Cont	73.40$_{\pm 0.01}$ (10)	13.63$_{\pm 0.02}$ (11)	36.34$_{\pm 0.02}$ (14)	14.46$_{\pm 0.03}$ (14)	1050.8$_{\pm 0.6}$ (11)	1031.3$_{\pm 0.9}$ (14)	10.1$_{\pm 0.3}$ (15)	8.1$_{\pm 0.4}$ (17)	13.25 (13)
	OUTFORMER	74.16$_{\pm 0.08}$ (7)	13.16$_{\pm 0.06}$ (10)	43.00$_{\pm 0.13}$ (3)	12.01$_{\pm 0.03}$ (6)	1062.9$_{\pm 1.4}$ (10)	1091.6$_{\pm 0.6}$ (5)	21.2$_{\pm 0.6}$ (4)	23.5$_{\pm 0.6}$ (3)	6.00 (5)
	FoMo-0D	72.65$_{\pm 0.04}$ (12)	13.87$_{\pm 0.08}$ (13)	42.70$_{\pm 0.13}$ (4)	12.26$_{\pm 0.10}$ (7)	1045.0$_{\pm 1.9}$ (13)	1085.1$_{\pm 2.4}$ (7)	20.8$_{\pm 0.8}$ (5)	25.3$_{\pm 0.6}$ (2)	7.88 (9)
	TabPFN-Extension	73.61$_{\pm 0.11}$ (8)	12.13$_{\pm 0.07}$ (6)	37.08$_{\pm 0.07}$ (13)	13.26$_{\pm 0.06}$ (11)	1090.1$_{\pm 1.7}$ (5)	1062.9$_{\pm 1.2}$ (11)	25.9$_{\pm 0.3}$ (3)	20.3$_{\pm 0.3}$ (5)	7.75 (8)
	TASKBRIDGE-TabPFNv3	75.91$_{\pm 0.06}$ (3)	12.05$_{\pm 0.06}$ (4)	**44.64**$_{\pm 0.17}$ **(1)**	11.91$_{\pm 0.12}$ (4)	1091.0$_{\pm 1.2}$ (3)	1095.1$_{\pm 2.7}$ (4)	26.3$_{\pm 0.5}$ (2)	**27.6**$_{\pm 0.8}$ **(1)**	2.75 (2)

dictive performance on TabArena (Erickson et al., 2025). We evaluate each variant on ODDBench against the same 30 TAD baselines. For each backbone, we perform the same one-time backbone-specific calibration of the framework-level hyperparameters for task selection and score aggregation, using the common configuration grid described in Appendix B.1.

As shown in Tables 11 and 12, TASKBRIDGE maintains strong overall performance when replacing the default TFM backbone with each alternative backbone, ranking fourth and second in terms of Total Rank, respectively. Notably, even when using the same TabPFN v3.0 backbone as TabPFN-Extension, TASKBRIDGE achieves better performance across all evaluation metrics, indicating that our framework harnesses the pretrained TFM more effectively for TAD. These results demonstrate that the core TAD mechanism of TASKBRIDGE—selecting suitable virtual supervised tasks and effectively aggregating task-wise scores—generalizes across different pretrained TFM backbones with strong capabilities for inferring task structure from supervised context and producing reliable predictive support.