TABMATE: A MULTI-AGENT FRAMEWORK FOR TABULAR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In tabular anomaly detection (AD), different anomaly mechanisms favor different detection strategies, making the complementary strengths of heterogeneous detectors valuable for reliable detection. However, this expertise can be lost when detector outputs are uniformly aggregated or reweighted toward consensus, particularly when useful specialists disagree with the majority. We introduce TabMATE, a **Tab**ular **M**ulti-**A**gent framework that organizes heterogeneous detection specialists into a coordinated **T**eam of **E**xperts through a shared score interface. The framework integrates large language model (LLM) agents as detectors spanning statistical, geometric, isolation-based, and foundation-model approaches, together with a learnable coordination layer that determines their contributions to the final prediction. Alongside the framework, we provide a theoretical analysis of the information limits of label-free coordination and a systematic evaluation on 45 ADBench datasets and four controlled anomaly types. The results show that three confirmed anomalies improve detection over uniform aggregation and the consensus-based coordination rules tested, with learned weights reflecting the specialization of the detector across types of anomalies. Further comparisons with alternative coordinators and standalone foundation models clarify the framework's practical scope, showing that stronger coordination does not necessarily yield a stronger predictor than an individual model under matched label budgets. TabMATE provides a modular foundation for integrating complementary detection expertise and systematically studying how specialist diversity and coordination shape multi-agent tabular AD.

1 INTRODUCTION

Tabular anomaly detection requires complementary expertise. An observation may be unusual because of a rare feature value, a sparse neighbourhood, or a broken dependency between otherwise ordinary values. These mechanisms favour different detection strategies, so an expert that is useful for one type of anomaly can be uninformative for another. Recent tabular foundation models expand the available capabilities through predictive and density-based inference (Hollmann et al., 2025; Qu et al., 2025; Shen et al., 2025; Ding et al., 2026; Wan et al., 2025). Context-aware benchmarks further show that anomalousness depends on what the task considers unusual (Yoon et al., 2026). The resulting challenge is to organize these different statistical views into a detector team that can use the expertise relevant to its target data.

We introduce **TabMATE**. This framework brings different heterogeneous detector agents into a shared decision process (Figure 1). Its agents specialize in marginal rarity, neighbourhood structure, projection and robust covariance, isolation, and foundation-model density estimation. Each agent assesses observations through its own scoring function and communicates normalized anomaly scores to a coordinator. Separating these specialist assessments from coordination allows the detector pool and the combination rule to be varied independently. The resulting weights and score contributions also make it possible to examine which experts the team uses, rather than evaluating only its final ranking.

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.

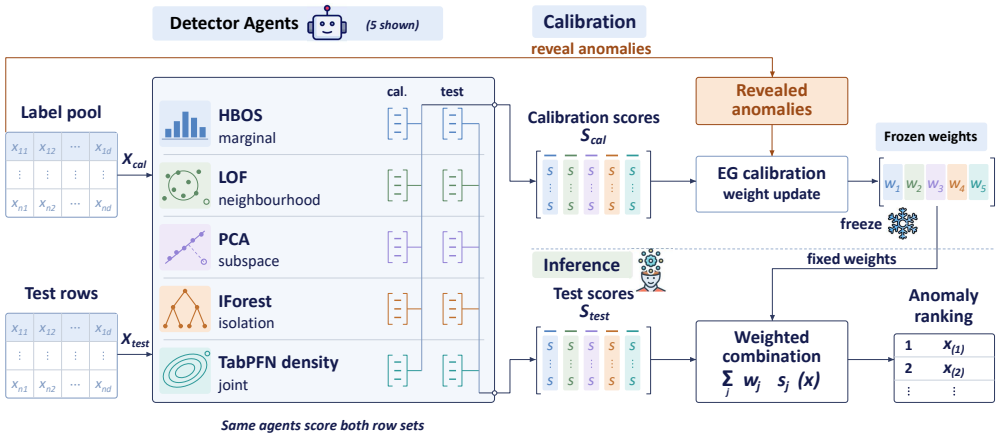


Figure 1: **TabMATE coordinates complementary detector experts.** Five (changable) detector agents are shown. The same agents score the training-side label pool and test rows. Revealed anomalies calibrate agent contributions through exponentiated gradient (EG); the resulting weights are frozen before combining test scores. Test scores bypass calibration.

TabMATE builds on outlier ensembles (Zimek et al., 2014) and relates to recent work on learned orchestration and structured multi-agent collaboration (Dang et al., 2025; Yun et al., 2026). Its focus is the collaboration among detection specialists themselves. This differs from assigning agents to construct a detection pipeline, as in AD-AGENT (Yang et al., 2025). The common score interface lets us study specialist composition, numerical coordination, and optional LLM coordination within the same detection setting. The primary detector agents are statistical models; LLMs are evaluated separately as coordinators and as reporters that interpret agent contributions.

Experiments on 45 ADBench datasets (Han et al., 2022) and four controlled anomaly types connect specialist capabilities to learned influence and team performance. Three confirmed anomalies improve mean ROC-AUC over uniform aggregation by 0.063 on real data and 0.109 on dependency anomalies, exceeding the tested label-free combiners in these settings. Learned weights track changes in specialist quality across anomaly types. We also compare alternative coordinators, realistic query accounting, stronger team members, and LLM interfaces. Matched-label comparisons with standalone foundation models test whether better coordination also yields a stronger predictor, and how this comparison changes with the feedback budget.

Our contributions are threefold:

A multi-agent framework for tabular anomaly detection. TabMATE connects heterogeneous detection specialists through a shared score interface and modular coordination. Exposes individual agent contributions and supports controlled comparisons of detector composition and coordination rules.

New coordination method guided by anomaly evidence. We calibrate agent influence that uses a few confirmed anomalies. We establish the limits of ordering agent quality from fixed label-free messages under unrestricted target labels and characterize a failure mode of disagreement-based updates.

An evaluation of complementary expertise and the utility of the team. Across real and controlled anomalies, we connect detection performance to specialist capabilities and learned weights. Comparisons with alternative coordinators, expanded pools, and individual foundation models identify where the framework benefits from cooperation and where a strong specialist remains preferable.

2 Related Work

Agent specialization and modular collaboration. AgentSquare standardizes planning, reasoning, tool-use, and memory interfaces (Shang et al., 2025); evolving orchestration learns agent selection and sequencing (Dang et al., 2025); and Graph-of-Agents selects and connects specialists (Yun et al.,

2026). AD-AGENT assigns agents to intent parsing, data preparation, model selection, and code generation (Yang et al., 2025). TabMATE places the detector specialists themselves in the team: each supplies numerical assessments to an interchangeable coordinator.

Outlier ensembles and detector selection. Heterogeneous outlier ensembles include selective combinations that reduce redundant or unreliable members (Zimek et al., 2014; Aggarwal & Sathe, 2017; Rayana & Akoglu, 2016; Zhao et al., 2019). SEAD adapts streaming ensembles using a score-derived proxy loss (Shah et al., 2025). MetaOD transfers detector selection from labelled auxiliary datasets (Zhao et al., 2021), while MetaEns learns marginal ensemble gains from such data (Phan et al., 2026). XGBOD learns directly from detector-score features (Zhao & Hryniewicki, 2018). TabMATE calibrates a fixed team with a few confirmed target anomalies and compares coordinators under matched scores and feedback.

Foundation models for tabular prediction and detection. TabPFN, TabICL, and TabDPT transfer predictive capabilities through in-context learning (Hollmann et al., 2025; Qu et al., 2025; Ma et al., 2025). FoMo-0D and OutFormer pretrain on synthetic detection tasks for inference without target anomaly labels (Shen et al., 2025; Ding et al., 2026). We combine a foundation-model density specialist with classical detectors, then add a predictive foundation model and compare the expanded team with direct prediction under matched revealed labels.

3 The TabMATE Framework

TabMATE organizes complementary detection specialists through a shared score interface and a modular coordinator (Figure 1). Given a training split, a calibration pool drawn from that split, and a test batch, the framework learns how much influence to assign each specialist and produces an anomaly ranking of the test rows.

3.1 Heterogeneous Detector Agents

The primary team contains ten agents with five families of inductive biases. HBOS, COPOD, and ECOD capture histogram or tail-based rarity; KNN, LOF, and CBLOF use neighbourhood or cluster structure; PCA and MCD measure projection residuals or deviations from robust covariance structure; Isolation Forest uses random isolation; and TabPFN density scores feature combinations through conditional density estimation (Hollmann et al., 2025). These views can disagree for useful reasons: a row may preserve ordinary feature values while breaking a dependency that a joint-density specialist can detect.

Each fitted agent i maps a row x to an anomaly score $s_i(x)$, oriented so that larger values indicate greater anomalousness. Agents are fitted or conditioned on training rows and remain fixed during coordination. We denote the number of valid agents in a split by N; failed or invalid scorers are omitted. Preprocessing and calibration-overlap details are given in Appendix A.

3.2 Shared Score Interface

To compare scores across agents, we replace their scales with normalized midranks. On a batch of n rows, the score matrix is

$$S_{ti} = \frac{\mathrm{midrank}(s_i(x_t))}{n}, \qquad S \in [0,1]^{n \times N}. \tag{1}$$

Ties receive average ranks. Calibration and test batches are ranked separately, yielding S^{cal} and S^{test}. These matrices remain fixed when comparing coordinators. Numerical coordinators read the matrices directly; an LLM coordinator reads reports summarizing agent families, scores, and agreement, together with a data card. The weight-based coordinators return $\alpha_i \geq 0$ with $\sum_{i=1}^{N} \alpha_i = 1$.

3.3 Coordination and Inference

Let P contain confirmed anomalies and Q contain comparison rows from the eligible calibration pool. Under *oracle-k*, P contains k supplied anomalies and Q contains the remaining eligible rows, including any hidden anomalies. Under *query-B*, labels are revealed for B randomly chosen rows:

Algorithm 1 TabMATE: calibrate the detector team, then freeze

Require: Fitted agents $\{s_i\}_{i=1}^N$, calibration pool, test batch, feedback budget
Ensure: Test anomaly scores $\widehat{s}$ and agent weights α
1: Compute S^{cal} and S^{test} using Eq. 1.
2: Construct P, Q under the oracle-k or query-B protocol.
3: Initialize $\alpha_i \leftarrow 1/N$ for all agents.
4: **if** $P \neq \emptyset$ **then**
5: Compute $\bar{\ell}_i$ and α_i using Eqs. 2–3.
6: **end if**
7: Freeze α; compute $\widehat{s} = S^{\text{test}}\alpha$.
8: **return** $\widehat{s}, \alpha$

discovered anomalies form P and revealed normals form Q. If no normal is revealed, the remaining eligible rows provide Q; if no anomaly is found, the coordinator retains uniform weights.

We instantiate label-driven coordination with exponentiated gradient (EG) (Freund & Schapire, 1997). For an anomaly–comparison pair (u, v), agent incurs $\ell_i(u, v) = [1 - (S^{\text{cal}}_{ui} - S^{\text{cal}}_{vi})]/2$. Its average loss over $P \times Q$ is

$$\bar{\ell}_i = \frac{1}{2}\left(1 - \widehat{\Delta}_i\right), \qquad \widehat{\Delta}_i = \frac{1}{|P|}\sum_{u\in P} S^{\text{cal}}_{ui} - \frac{1}{|Q|}\sum_{v\in Q} S^{\text{cal}}_{vi}. \tag{2}$$

Thus, an agent gains influence by separating confirmed anomalies from comparison rows. Starting uniformly and processing all $T = |P||Q|$ pairs gives the exact terminal weights

$$\alpha_i = \frac{\exp\left(-\eta T \bar{\ell}_i\right)}{\sum_{j=1}^N \exp\left(-\eta T \bar{\ell}_j\right)}, \qquad \eta = \sqrt{\frac{8\log N}{T}}. \tag{3}$$

Only two column means are needed; explicit pair enumeration is unnecessary. The pairs reuse labels, so T counts update terms rather than independent annotations. The standard online regret guarantee does not imply a test-AUC guarantee for these terminal weights (Appendix B.2).

We freeze α after calibration and combine test scores as

$$\widehat{s}(x_t) = \sum_{i=1}^N \alpha_i S^{\text{test}}_{ti}. \tag{4}$$

Weights adapt at the dataset level and are shared across its test rows. Algorithm 1 summarizes this process. The interface also supports alternative weighting rules and additional detector agents.

3.4 Task Feedback

A label-free coordinator chooses $\alpha = W(S, Z)$ from score messages and fixed side information Z, such as unlabelled features or a pretrained prior. Holding these inputs fixed exposes a limit on identifying useful expertise.

Theorem 1 (Relative quality under fixed messages)**.** *Fix a target-label-free coordinator W and two agents with distinct midrank vectors on $n \geq 2$ rows. For every $m \in \{1, \ldots, n-1\}$, there exists a complete binary labelling with exactly m anomalies under which the higher-AUC agent receives no more weight than the lower-AUC agent.*

The difference between the two rank vectors sums to zero. Labelling its largest or smallest entries as anomalies reverses relative AUC while leaving the coordinator's inputs unchanged (Appendix B.1). The result concerns unrestricted target labelings; valid domain assumptions can exclude these alternatives.

Agreement penalties create a related failure mode. An EG update decreases an agent's weight whenever its loss exceeds every other agent's loss and its current weight lies in $(0, 1)$. A correct specialist can therefore lose influence when disagreement itself is penalized (Appendix B.3). Label-driven coordination instead evaluates each specialist against revealed task outcomes.

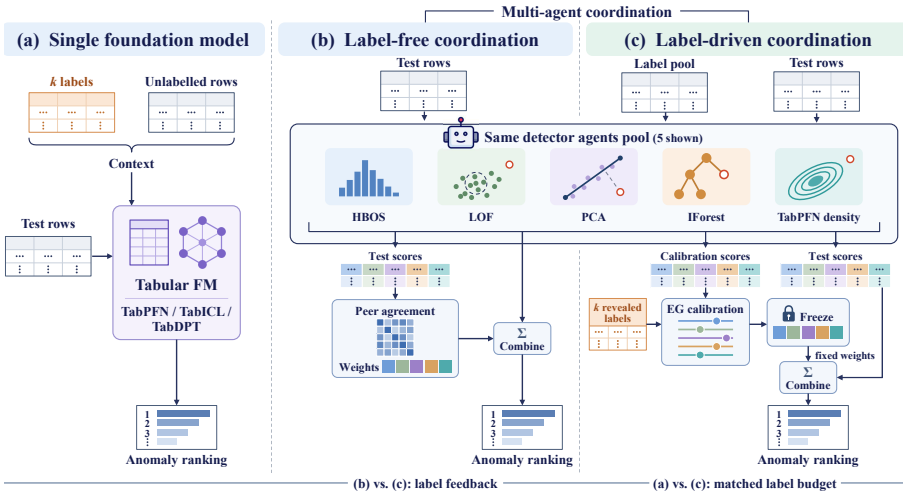


Figure 2: **Comparisons with other AD Method/Model.** (a) A single tabular foundation model uses revealed labels and unlabelled rows as context. (b) A label-free coordinator combines the detector team's score messages; peer agreement is one example. (c) The same detector agents score a training-side label pool, allowing EG to learn and freeze weights before combining test scores.

4 Experiments

We test which detector specialists contribute useful information, how coordination recovers it, and when a team improves on an individual model. Figure 2 contrasts coordination under different feedback with direct prediction at matched revealed-label budgets.

4.1 Experimental Setup

We use 45 ADBench datasets (Han et al., 2022) and four generated anomaly types: local, clustered, global, and dependency. Their generators respectively inflate mixture covariance, shift component means, sample an expanded feature range, and break feature dependencies. These controlled settings let us compare the same detector families under different anomaly structures. Each dataset is capped at 10^4 rows and divided equally into training and test halves with stratification; calibration and test batches contain at most 2000 rows each. The primary evaluation uses two split seeds and five calibration draws per split. Two dependency-generation runs are unavailable, leaving 358 successful type–seed settings. ROC-AUC is averaged across seeds and draws within each dataset, then equally across datasets. Reported 95% confidence intervals use 5000 bootstrap resamples of dataset means.

The primary pool contains the ten agents in Section 3.1; uniform averaging is the reference for reported gains. Label-free alternatives include robust aggregation, agreement-based weighting and pruning, a spectral rule inspired by Parisi et al. (2014), and consensus EG. Label-driven alternatives use the same revealed rows as TabMATE. Oracle-k supplies k confirmed anomalies, whereas query-B reveals the labels of B randomly selected eligible rows. The former measures the value of available anomaly examples; the latter also accounts for discovering them. Budget eligibility determines dataset coverage, which is reported with each comparison. External-model and expanded-pool comparisons use the first two calibration draws, and LLM coordination uses the first oracle-3 draw. Their matched EG references are therefore computed on those same draws.

The evaluation uses transductive preprocessing: feature scaling and PCA for wide real datasets precede splitting, and rank normalization uses complete score batches. Classical calibration scores are in-sample. Stratified candidate sampling and a fallback for density-context overlap also use hidden labels during benchmark preparation. The reported budgets count labels subsequently revealed to the coordinator; test labels are not used to fit its weights. All coordination weights are frozen before test prediction. We report an overlap sensitivity analysis below and provide full protocol details in the appendix.

Table 1: **Detection performance with the same specialist pool.** Uniform reports mean ROC-AUC; other rows report gains over uniform. Label-free rules receive no revealed labels; EG receives three confirmed anomalies. The best-agent gap uses retrospective test-label selection.

Method	Real	Local	Clustered	Global	Dependency
Uniform (AUC)	0.763	0.904	0.929	0.997	0.826
Median	+0.002	−0.005	+0.024	0.000	−0.006
Average of maxima (AOM)	+0.008	+0.002	+0.022	0.000	+0.005
Consensus pruning	−0.007	−0.012	+0.025	−0.001	−0.074
Agreement weighting	−0.003	−0.008	+0.020	−0.001	−0.045
Spectral	0.000	−0.002	+0.010	0.000	−0.022
Consensus EG	−0.015	−0.019	+0.024	−0.001	−0.118
Model centrality	−0.005	−0.017	+0.016	−0.001	−0.031
HITS	−0.006	−0.017	+0.019	−0.001	−0.031
TabMATE (EG)	+0.063	+0.014	+0.047	−0.003	+0.109
Best-agent gap	+0.073	+0.040	+0.059	+0.002	+0.131

4.2 Overall Detection Performance

TabMATE improves the fixed detector team on real, local, clustered, and dependency anomalies (Table 1). With three confirmed anomalies, mean real-data ROC-AUC rises from 0.826 to 0.806, a gain of 0.063 with a 95% bootstrap interval of $[0.023, 0.066]$. The gain reaches 0.109 on dependency anomalies, where detecting an unusual combination of otherwise ordinary values requires expertise that several marginal scorers lack. These gains recover approximately 86% and 83% of the respective best-agent gaps. The best-agent reference selects a member retrospectively using test labels; it measures available individual expertise and is not an upper bound on ensemble performance.

Among the tested label-free rules, average of maxima (AOM) has the largest real-data gain, 0.008. Several agreement-based rules help clustered anomalies but lose performance on dependency anomalies. Consensus EG, for example, gains 0.024 on clustered anomalies and loses 0.118 on dependency anomalies. Because the detector scores are held fixed, these differences show that the coordination rule determines how much of the pool's expertise contributes to the final ranking. They also motivate inspecting individual agents rather than treating diversity alone as sufficient. Reweighting is not uniformly beneficial: global anomalies are already nearly solved by uniform averaging at 0.997 AUC, and EG lowers that score by 0.003.

4.3 Which Coordination Rule Works?

The shared score interface allows the coordinator to change while the specialists and revealed rows stay fixed. Table 2 compares selection, weighting, and stacking rules. Selecting the estimated best agent and AUC reweighting estimate quality against the remaining eligible rows, treated as normal. Thompson and UCB weighting use pairwise ranking outcomes, while logistic and boosted stacking learn from agent score vectors. These latter rules use the same positive and comparison sets as EG. The mirrored-expert variant additionally includes sign-reversed agent scores, allowing EG to exploit a detector whose orientation is unhelpful for the current task.

Several rules benefit from the feedback. At three confirmed anomalies, selection gains 0.041, compared with 0.063 for EG. TabMATE has the largest mean gains at three and ten anomalies, its three-anomaly advantage over other methods is not significant under an exploratory dataset-level Wilcoxon signed-rank test ($p = 0.16$). Its signed coefficients permit combinations beyond nonnegative simplex weighting. These comparisons support learning from anomaly feedback without identifying EG as the best supervised combiner. We use it as the default because its nonnegative, normalized weights provide a direct account of each specialist's influence. TabMATE makes this choice explicit and replaceable, so improved coordination can be evaluated without rebuilding the detector pool.

Table 2: **Coordination rules at matched feedback budgets.** Real-data ROC-AUC gains over uniform. Zero-anomaly query draws use uniform fallback. Bold marks the largest mean; coverage depends on budget eligibility.

	Oracle-k			Query-B	
Coordinator	1	3	10	30	100
Select estimated best	+0.010	+0.041	+0.059	+0.035	+0.049
AUC reweighting	+0.022	+0.031	+0.042	+0.026	+0.034
Thompson weighting	+0.020	+0.031	+0.042	+0.027	+0.034
UCB weighting	+0.020	+0.027	+0.033	+0.015	+0.026
Logistic stacking	+0.010	+0.056	+0.103	+0.050	+0.083
Boosted stacking	−0.117	−0.058	+0.042	−0.181	+0.060
TabMATE (EG)	**+0.030**	**+0.063**	**+0.148**	**+0.060**	**+0.095**
EG, mirrored experts	−0.020	+0.132	+0.068	+0.096	+0.064
Datasets	45	45	38	45	41
Mean anomalies revealed	1.0	3.0	10.0	3.6	11.8

Table 3: **Acquisition and calibration-pool controls.** (a) Query-30 includes all 180 draws; 51 zero-anomaly draws use uniform fallback for every method. EG-11 adds TabICL. (b) Oracle-3 EG gains after excluding flagged density-context overlap; classical calibration remains in-sample. Parentheses give dataset counts.

(a) All-query performance

Method	AUC	Gain
EG-10	0.899	+0.136
EG-11	0.926	+0.163
TabICL	0.850	+0.087

(b) Overlap sensitivity

Type	All (45)	Unflagged (29)
Real	+0.063	+0.040
Clustered	+0.047	+0.047
Dependency	+0.109	+0.099

4.4 Label Budgets and Query Protocols

More confirmed anomalies generally improve calibration, but acquiring them has a separate cost. In the primary five-draw evaluation, 30 random queries reveal 3.6 anomalies on average and find none in 113 of 450 draws (25.1%). We retain those draws with uniform predictions. The resulting EG gain is 0.060, compared with 0.063 when three anomalies are supplied directly (Table 2). These are results on the prepared candidate pools, whose anomaly prevalence can be raised by capped sampling; they should not be interpreted as discovery rates for uncapped deployment data. Confirmed examples and annotation queries measure distinct resources.

The strong-model study uses two draws per split and therefore has a different set of query outcomes. To evaluate a complete query policy, we combine its successful query-30 runs with the same uniform fallback for all methods on the 51 of 180 draws that find no anomaly. This includes draws on which a label-conditioned model could not be fitted, instead of evaluating only successful discovery. Across all 45 datasets, EG-10, EG-11, and TabICL gain 0.136, 0.163, and 0.087, respectively (Table 3a). Here EG-11 adds TabICL as an eleventh agent, as examined in Section 4.6. The individual model's advantage therefore remains under this explicitly specified all-query policy.

Calibration-pool preparation is another relevant condition. Excluding the 16 datasets flagged for density-context overlap reduces EG's real-data gain from 0.063 to 0.040 and its dependency gain from 0.109 to 0.099, while the clustered gain remains 0.047 (Table 3b). The improvements persist on the unflagged subset, although this comparison also changes dataset composition. It does not make classical calibration out-of-sample or remove the transductive preprocessing described above.

4.5 Complementary Expertise and Agent Contributions

The agents contribute different statistical views, and their usefulness changes with anomaly structure (Figure 3a). LOF attains 0.923 AUC on local anomalies but only 0.543 on clustered anomalies, where PCA reaches 0.978. TabPFN density reaches 0.953 on dependency anomalies and is the best agent on 80% of datasets, yet reaches only 0.806 on clustered anomalies. These reversals show why the framework retains multiple specialist families. They also explain why adding an agent is not

sufficient: including density in uniform aggregation improves dependency detection by 0.029 but reduces clustered detection by 0.007. The same specialist can supply missing information in one setting and dilute more relevant information in another.

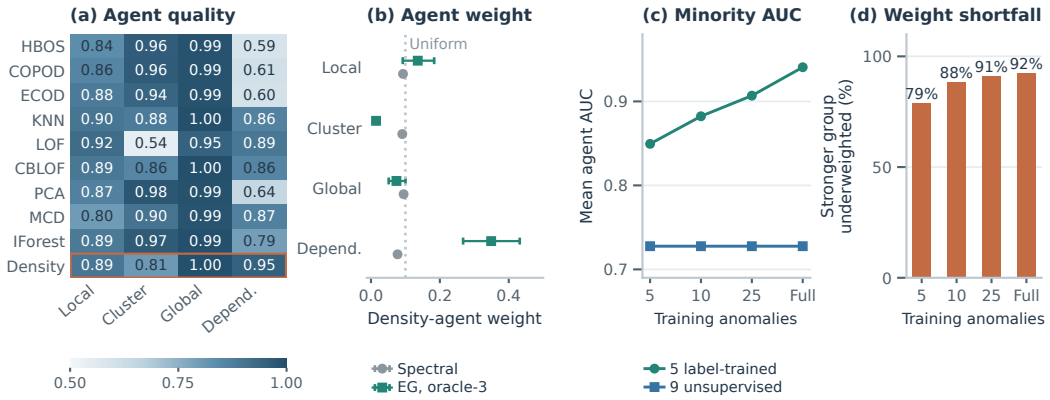


Figure 3: **Specialist quality and influence.** (a) Agent AUC by anomaly type. (b) TabPFN-density weights under full-covariance spectral coordination and oracle-3 EG; dotted line: uniform weight (0.1); error bars: 95% dataset-bootstrap intervals. (c) Mean AUC of five label-trained and nine unsupervised agents. (d) Fraction of runs in which the group with higher mean AUC receives less than its count-proportional spectral weight. Panels (c–d) use 31 datasets and five seeds. Their labels train the agents, separately from the three anomalies used to calibrate EG in (b).

TabMATE's learned influence reflects these differences (Figure 3b). EG assigns the density agent weight 0.349 on dependency anomalies and 0.015 on clustered anomalies. The full-covariance spectral diagnostic gives similar weights, 0.077 and 0.091; its diagonal treatment differs from the primary spectral coordinator (Appendix C). The correlation between EG weights and individual-agent AUC, computed within runs and then averaged equally across datasets, reaches 0.80 on dependency anomalies and 0.69 on clustered anomalies. A separate diagnostic examines whether improved agent capability necessarily leads to greater influence. We add five label-trained agents to nine fixed unsupervised agents and increase the supervision available to the trained group on 31 datasets and five seeds (Figure 3). Its mean AUC rises from 0.849 to 0.941, yet spectral coordination assigns the better group less than its count-proportional weight in 79–92% of runs. These labels train the agents themselves; they are distinct from the small feedback budget used to calibrate TabMATE's coordinator.

The classical-agent diagnostic gives a related explanation. On 38 datasets and five seeds, the leading spectral vector correlates with mean peer agreement at a median $\rho = 0.983$, but with agent AUC at only 0.283. A dominant common score pattern can therefore describe agreement without reliably identifying which detectors are useful. These observations support the design choice to calibrate influence using task feedback. They do not imply that every label-free rule fails, nor do the generated anomaly types directly test the fixed-message relabelling construction of Theorem 1: the generators change the observations and hence the agents' scores.

4.6 Comparison with Individual Models and Expanded Teams

Table 4 compares direct prediction and an expanded team under matched revealed labels within each block; the blocks use separate runs. In the direct comparison, EG-10 leads at one and ten anomalies (+0.057 and +0.188), and is within 0.001 of TabICL at three. With TabICL added, EG-11 leads at one and three anomalies (+0.061 and +0.109), but at ten TabICL alone reaches +0.161 versus +0.129 for EG-11. Thus, the team's advantage over an individual model depends on the feedback budget. EG timing covers coordination after scoring; external-model timing includes fitting and scoring.

Table 4: **Direct prediction and a stronger team.** Gains over uniform-10; budgets $1, 3, 10$ cover $45, 45, 38$ datasets. Labels match within each block; direct and expanded-pool results use separate matched runs. Times are per run on H200, including fitting and scoring for external models; EG timing excludes pool scoring. TabICL uses held-out calibration in the expanded pool (Appendix F).

Method	$k = 1$	$k = 3$	$k = 10$	s/run
Direct prediction with matched revealed labels				
TabMATE (EG-10)	$+0.057$	$+0.101$	$+0.188$	< 0.1
TabPFN classifier (Hollmann et al., 2025)	$+0.042$	$+0.095$	$+0.147$	0.6
TabICL (Qu et al., 2025)	$+0.052$	$+0.102$	$+0.162$	2.7
TabDPT (Ma et al., 2025)	$+0.054$	$+0.099$	$+0.159$	3.6
REPEN (Pang et al., 2018)	$+0.020$	$+0.048$	$+0.072$	40.3
DevNet (Pang et al., 2019)	-0.070	$+0.007$	$+0.074$	10.8
PReNet (Pang et al., 2023)	-0.069	$+0.005$	$+0.069$	9.8
DeepSAD (Ruff et al., 2020)	-0.031	$+0.001$	$+0.056$	3.6
FEAWAD (Zhou et al., 2022)	-0.103	-0.010	$+0.053$	23.1
XGBOD (Zhao & Hryniewicki, 2018)	-0.094	$+0.003$	$+0.088$	1.6
XGBoost on features	-0.077	-0.029	$+0.075$	1.6
FT-Transformer (Gorishniy et al., 2021)	-0.015	$+0.029$	$+0.077$	48.7
ResNet (rtdl) (Gorishniy et al., 2021)	-0.185	-0.149	-0.088	8.0
TabM (Gorishniy et al., 2025)	-0.144	-0.031	$+0.075$	1.5
Expanded pool: TabICL joins as an eleventh agent				
TabICL alone	$+0.053$	$+0.101$	$+0.161$	—
Uniform-11	$+0.011$	$+0.016$	$+0.024$	—
Select estimated best, 11	$+0.014$	$+0.062$	$+0.129$	—
TabMATE (EG-11)	$+0.061$	$+0.109$	$+0.129$	—
Logistic stacking, 11	-0.016	$+0.031$	$+0.101$	—

5 Discussion and Limitations

TabMATE exposes individual detector contributions and separates specialist composition from coordination. Task feedback improves the use of complementary expertise, but superiority to a strong individual model depends on the detector pool and label budget. These factors must therefore be evaluated together.

Practical utility depends on the resources needed to obtain labels and produce specialist scores. Three confirmed anomalies and three annotation queries are different budgets. The query experiments account for draws that find no anomaly, but their candidate pools still follow benchmark preparation that can change prevalence. Computation is also uneven across agents: on the reported H200 setup, pool scoring takes 84.3 seconds, of which the density agent accounts for 82.8 seconds; EG adds less than 0.1 seconds. TabICL's reported fitting-and-scoring run takes 2.7 seconds. A lightweight coordination rule therefore does not imply an inexpensive end-to-end team. Matching revealed labels also leaves pretraining resources and model computation unmatched.

6 Conclusion

We introduced TabMATE, a multi-agent framework that organizes heterogeneous tabular anomaly detectors into a team of complementary specialists. Its shared score interface and modular coordination layer make individual agent contributions explicit and allow their influence to be calibrated with limited feedback. Our theoretical analysis explains why agreement can overlook useful expertise. Evaluation on 45 ADBench datasets and four controlled anomaly types connects detector specialization to learned influence and team performance. With three confirmed anomalies, TabMATE improves mean ROC-AUC on real data over uniform aggregation and the tested consensus-based rules, while controlled experiments show how calibration adapts agent weights to different anomaly mechanisms. TabMATE provides a modular basis for integrating new detection specialists and developing coordination strategies that make effective use of their complementary capabilities.

AI USE STATEMENT

We used generative AI tools for literature review and polishing the final writing. We also used AI assistance to revise scientific illustrations. The authors directed the revisions and retain responsibility for the correctness, originality, and final presentation of the work, including all scientific claims, analyses, text, and figures. The use of LLMs as experimental coordinators and reporters, including their inputs and evaluation protocols, is documented separately in Appendix G.

ETHICS STATEMENT

This study evaluates anomaly detection using existing ADBench datasets and statistically generated anomalies. An anomaly score indicates deviation from learned patterns and does not establish wrongdoing or justify an adverse decision about an individual. Applications involving people require attention to false positives, differences in error rates across groups, and the consequences of incorrect alerts.

REPRODUCIBILITY STATEMENT

Section 3 specifies the detector interface and coordination procedure. Appendix A documents dataset preparation, detector configurations, random seeds, synthetic anomaly generation, and label-budget protocols. Appendix B provides theoretical assumptions and proofs. Appendices C and F describe alternative coordinators, external models, and supplementary controls. Appendix G documents the LLM prompts, model settings, and response processing, while Appendix H specifies dataset coverage, aggregation, and statistical reporting. Implementation and record-availability limitations are stated alongside the relevant experimental protocols.

REFERENCES

Charu C. Aggarwal and Saket Sathe. *Outlier Ensembles: An Introduction*. Springer, 2017.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. Multi-agent collaboration via evolving orchestration. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5502. URL https://arxiv.org/abs/2505.19591.

A. Philip Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1): 20–28, 1979. doi: 10.2307/2346806. URL https://doi.org/10.2307/2346806.

Xueying Ding, Haomin Wen, Simon Klüttermann, and Leman Akoglu. From zero to hero: Advancing zero-shot foundation models for tabular outlier detection. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.03018.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Nicolas Goix. How to evaluate the quality of unsupervised anomaly detection algorithms? In *ICML Workshop on Anomaly Detection*, 2016.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, 2021.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.24210.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. ADBench: Anomaly detection benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32142–32159, 2022. URL https://arxiv.org/abs/2206.09426.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Ariel Jaffe, Boaz Nadler, and Yuval Kluger. Estimating the accuracies of multiple classifiers without labeled data. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 407–415. PMLR, 2015. URL https://proceedings.mlr.press/v38/jaffe15.html.

Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 157–166, 2005. doi: 10.1145/1081870.1081891. URL https://doi.org/10.1145/1081870.1081891.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2410.18164.

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. *ACM SIGKDD Explorations Newsletter*, 25(1):19–35, 2023. doi: 10.1145/3606274.3606277. URL https://doi.org/10.1145/3606274.3606277.

Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2041–2050, 2018. doi: 10.1145/3219819.3220042. URL https://doi.org/10.1145/3219819.3220042.

Guansong Pang, Chunhua Shen, and Anton van den Hengel. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 353–362, 2019. doi: 10.1145/3292500.3330871. URL https://doi.org/10.1145/3292500.3330871.

Guansong Pang, Chunhua Shen, Huidong Jin, and Anton van den Hengel. Deep weakly-supervised anomaly detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1795–1807, 2023. doi: 10.1145/3580305.3599302. URL https://doi.org/10.1145/3580305.3599302.

Fabio Parisi, Francesco Strino, Boaz Nadler, and Yuval Kluger. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258, 2014.

Hong-Phuc Phan, Tuan-Anh Vu, Tung Kieu, Son Ha Xuan, Bin Yang, and Christian S. Jensen. Automatic unsupervised ensemble outlier model selection. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.16567. Extended version available as arXiv:2605.16567.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847. PMLR, 2025. URL https://proceedings.mlr.press/v267/qu25d.html.

Shebuti Rayana and Leman Akoglu. Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data*, 10(4):1–33, 2016. doi: 10.1145/2890508. URL https://doi.org/10.1145/2890508.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

Erich Schubert, Remigius Wojdanowski, Arthur Zimek, and Hans-Peter Kriegel. On evaluation of outlier rankings and outlier scores. In *SIAM International Conference on Data Mining (SDM)*, pp. 1047–1058, 2012. doi: 10.1137/1.9781611972825.90. URL https://doi.org/10.1137/1.9781611972825.90.

Saumya Gaurang Shah, Abishek Sankararaman, Balakrishnan Murali Narayanaswamy, and Vikramank Singh. SEAD: Unsupervised ensemble of streaming anomaly detectors. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54167–54185. PMLR, 2025. URL https://proceedings.mlr.press/v267/shah25c.html.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. AgentSquare: Automatic LLM agent search in modular design space. In *International Conference on Learning Representations*, 2025. URL https://iclr.cc/virtual/2025/poster/32059.

Yuchen Shen, Haomin Wen, and Leman Akoglu. FoMo-0D: A foundation model for zero-shot tabular outlier detection. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=XCQzwpR9jE.

Guangya Wan, Yuqi Wu, Hao Wang, Shengming Zhao, Jie Chen, and Sheng Li. Derailer-Rerailer: Adaptive verification for efficient and reliable language model reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 327–348, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.18. URL https://aclanthology.org/2025.findings-acl.18/.

Tiankai Yang, Junjun Liu, Michael Siu, Jiahang Wang, Zhuangzhuang Qian, Chanjuan Song, Cheng Cheng, Xiyang Hu, and Yue Zhao. AD-AGENT: A multi-agent framework for end-to-end anomaly detection. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 191–205, 2025. doi: 10.18653/v1/2025.findings-ijcnlp.11. URL https://aclanthology.org/2025.findings-ijcnlp.11/.

Sanghyu Yoon, Dongmin Kim, Suhee Yoon, Ye Seul Sim, Seungdong Yoa, Hye-Seung Cho, Soonyoung Lee, Hankook Lee, and Woohyung Lim. ReTabAD: A benchmark for restoring semantic context in tabular anomaly detection. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UFwgg44VZq.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Zou, Guohao Li, and Tianlong Chen. Graph-of-Agents: A graph-based framework for multi-agent LLM collaboration. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=34cANdsHKV.

Yue Zhao and Maciej K. Hryniewicki. XGBOD: Improving supervised outlier detection with unsupervised representation learning. In *International Joint Conference on Neural Networks (IJCNN)*, 2018.

Yue Zhao, Zain Nasrullah, Maciej K Hryniewicki, and Zheng Li. LSCP: Locally selective combination in parallel outlier ensembles. In *SIAM International Conference on Data Mining (SDM)*, pp. 585–593, 2019.

Yue Zhao, Ryan A. Rossi, and Leman Akoglu. Automatic unsupervised outlier model selection. In *Advances in Neural Information Processing Systems*, 2021.

Yingjie Zhou, Xucheng Song, Yanru Zhang, Fanxing Liu, Ce Zhu, and Lingqiao Liu. Feature encoding with autoencoders for weakly supervised anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2454–2465, 2022. doi: 10.1109/TNNLS.2021.3086137. URL https://doi.org/10.1109/TNNLS.2021.3086137.

Arthur Zimek, Ricardo J. G. B. Campello, and Jörg Sander. Ensembles for unsupervised outlier detection: Challenges and research questions. *ACM SIGKDD Explorations Newsletter*, 15(1):11–22, 2014. doi: 10.1145/2594473.2594476. URL https://www.kdd.org/exploration_files/V15-01-02-Zimek.pdf.

A Implementation and Evaluation Protocol

A.1 Data Preparation and Score Generation

The primary experiments use 45 eligible datasets from the classical ADBench collection (Han et al., 2022). Each dataset is capped at 10,000 rows by class-stratified subsampling. The subsample approximately preserves the original prevalence, subject to retaining at least 20 anomalies when that many are available. Datasets with fewer than ten anomalies, or fewer than five anomalies in the test half, are omitted. Features are standardized and constant columns are removed. Real datasets wider than 64 features are reduced to 64 principal components. A stratified 50/50 split supplies the training and test halves; test batches are capped at 2000 rows. The primary experiments use split seeds zero and one. Sampling before the split uses random seed $700 + \text{seed}$.

Classical detector agents are fitted on the training half and score both the test batch and a stratified calibration pool of at most 2000 training rows. TabPFN density instead receives a uniformly sampled context of at most 1000 training rows. Every classical agent uses its anomaly-oriented decision function. It is retained only if fitting succeeds, its calibration and test scores are finite, and its test scores have standard deviation greater than 10^{-12}. Score generation requires at least six valid classical agents and a successful density fit. The reported pool has ten nominal members; N in the equations denotes the valid agents in that split.

Table 5 lists the settings explicitly supplied to the primary detector constructors; other options use the installed PyOD defaults. The experiment records do not pin every transitive dependency version, so these settings are not a complete environment specification.

Table 5: Primary detector-agent settings. The last column lists explicitly configured options; omitted options use library defaults.

Agent	Statistical view	Explicit configuration
HBOS	Feature histograms	PyOD HBOS constructor
COPOD	Copula-inspired tail scores	PyOD COPOD constructor
ECOD	Empirical distribution tails	PyOD ECOD constructor
KNN	Neighbour distance	20 neighbours; parallel scoring
LOF	Relative local density	20 neighbours; parallel scoring
CBLOF	Cluster structure	Random state 0
PCA	Projection-based score	PyOD PCA constructor
MCD	Robust covariance	Random state 0
IForest	Random isolation	Random state 0; parallel scoring
TabPFN density	In-context joint density	TabPFN v2 classifier and regressor; one estimator each; 1000 context rows; three feature orders

Calibration and test columns are normalized separately by empirical midrank divided by batch size, as in Eq. 1. Coordinator comparisons on a fixed pool reuse these matrices. The pipeline is transductive: feature standardization and PCA for wide real datasets are fitted before splitting, and test ranks use the full test batch. Freezing weights before test-label evaluation therefore does not make the full pipeline inductive. Test labels are reserved for evaluation; hidden labels also enter the benchmark preparation steps described below.

A.2 Calibration Candidates and Label Accounting

The eligible calibration candidates initially exclude rows in the density agent's context. If fewer than 50 rows or no anomaly remain, the entire calibration pool is used and the run is flagged. This fallback occurs on 16 of the 45 real datasets. Stratification and the positive-availability check use hidden labels during benchmark preparation; they are not a deployable label-blind query policy. Excluding density-context overlap also does not make classical calibration scores out of sample, because their agents were fitted on the training half that supplies those rows. Table 3 reports the sensitivity analysis after excluding flagged datasets.

For oracle-k, the coordinator receives k confirmed anomalies from the eligible candidates, with $k \in \{1, 3, 10\}$. For query-B, it receives the labels of B uniformly sampled candidates, with

$B \in \{10, 30, 100\}$. All methods use the same revealed row indices in a given draw. A shared random generator with seed $1000\,\mathrm{seed} + \mathrm{draw}$ generates oracle draws in increasing k order followed by query draws in increasing B order; methods use separate internal randomness. There are five draws per seed in the primary coordinator study and two in the external-model and expanded-pool comparisons. A budget is included only when the eligible pool contains the requested number of positives or queried rows.

For EG, revealed anomalies form P. In query mode, revealed normals form Q when any are found; otherwise the remaining candidates outside P provide Q. Oracle mode always uses the latter comparison set, including any hidden anomalies. If no anomaly is discovered, the fixed-pool rules return the uniform score. The evaluated update cases have nonempty Q. In contrast, estimated-best selection and AUC reweighting estimate each agent's AUC against *all* eligible candidates outside P, including in query mode. Thus, matched revealed labels do not mean that every rule uses the same comparison rows.

A.3 TabPFN Density and Synthetic Anomalies

The density agent computes chain-rule negative log likelihood with TabPFN v2 conditional predictors. Continuous-feature terms use the predictive criterion's negative-log-likelihood output; categorical terms use $-\log p$. Each contribution is clipped to $[-30, 30]$, and scores are averaged over three feature orders. This modification addresses inverse-density behavior observed with the TabPFN 2.2.x and extension combination used in the experiment; it does not characterize all model releases.

The four controlled anomaly types adapt ADBench's realistic-synthetic construction. Local, clustered, and global settings fit Gaussian mixtures to normal rows, selecting one to nine components by BIC. Normal observations are sampled from the fitted mixture. Local anomalies inflate component covariances by five; clustered anomalies multiply component means by five. For global anomalies, each feature is drawn uniformly between 1.1 times the minimum and 1.1 times the maximum of the generated normal sample. Dependency data use a fitted center vine copula for normal rows and independent Gaussian-KDE marginal draws for anomalies.

Generation operates on raw features before standardization. This order preserves the intended clustered mean shift, which could collapse after centering. Wide inputs use a seeded subset of at most 64 raw features; the dependency generator may further select 50. Constant features are removed, and copula fitting uses at most 5000 normal rows. Generated class counts match the original sampled class counts. Nonfinite generated values are repaired and clipped using the observed normal-feature ranges. There are 358 successful type–seed combinations rather than 360; every base dataset retains at least one successful seed for each type. Since the generator changes observations and their scores, this study measures specialization across data-generating settings, not the fixed-message relabelling construction of Theorem 1.

A.4 End-to-End Algorithm

Algorithm 2 gives the inference procedure after the detector pool is fitted. It learns one dataset-level weight vector shared by all test rows. The contribution $\alpha_i S_{ti}^{\mathrm{test}}$ is a component of the combined score, not a causal attribution.

B Theoretical Details

B.1 Proof of Theorem 1

Proof. Let $r_i, r_j \in \mathbb{R}^n$ be the two midrank vectors, and write $d = r_i - r_j$. Midranks sum to $n(n+1)/2$, so $\sum_{t=1}^{n} d_t = 0$. Distinct rank vectors imply $d \neq 0$. For any $1 \leq m < n$, the sum of the largest m entries of d is strictly positive, and the sum of the smallest m entries is strictly negative. Otherwise, the zero overall sum and the ordering of the entries would force $d = 0$.

For a positive set A of size m, empirical AUC with half credit for ties is

Algorithm 2 TabMATE: calibrate detector contributions, then freeze

Require: Fitted agents $\{s_i\}_{i=1}^N$, calibration pool, density-context indices, test batch, feedback protocol and budget
Ensure: Test scores $\widehat{s}$ and detector weights α
1: Score calibration and test rows with each valid agent.
2: Rank-normalize each column within its own batch to obtain S^{cal} and S^{test}.
3: Construct eligible calibration candidates as described in Appendix A.
4: Reveal oracle-k anomalies or query-B row labels; construct P and Q under the selected protocol.
5: Initialize $\alpha_i \leftarrow 1/N$ for all agents.
6: **if** $P \neq \emptyset$ **then**
7: Compute $\widehat{\Delta}_i$ and $\bar{\ell}_i$ by Eq. 2; set $T \leftarrow |P||Q|$.
8: Set $\eta \leftarrow \sqrt{8 \log N / T}$ and $z_i \leftarrow -\eta T \bar{\ell}_i$.
9: Set $\alpha_i \leftarrow \exp(z_i - \max_j z_j) / \sum_j \exp(z_j - \max_h z_h)$.
10: **end if**
11: Freeze α and compute $\widehat{s} \leftarrow S^{\mathrm{test}} \alpha$.
12: **return** $\widehat{s}, \alpha$.

$$\mathrm{AUC}_i(A) = \frac{\sum_{t \in A} r_{ti} - m(m+1)/2}{m(n-m)}. \tag{5}$$

Consequently, the difference in the two AUCs has the sign of $\sum_{t \in A} d_t$. Choosing the largest m entries as anomalies makes agent i better; choosing the smallest m makes agent j better. All remaining rows are normal in both constructions, and (S, Z) is unchanged. If $W_i \leq W_j$, choose the first labelling; otherwise choose the second. The better agent receives no more weight in either case. □

The statement conditions on a fixed observed instance and allows unrestricted target labels. It establishes a boundary on identifying relative quality from fixed messages; domain assumptions can rule out the constructed labelings. For a randomized coordinator, the argument applies to weights averaged over its internal randomness, conditional on the fixed inputs.

B.2 Exact EG Computation and the Scope of Its Guarantee

An EG update starts from $\alpha_i^{(1)} = 1/N$ and applies $\alpha_i^{(t+1)} \propto \alpha_i^{(t)} \exp(-\eta \ell_i^{(t)})$. Since the pairwise loss is linear in the two scores,

$$\sum_{(u,v) \in P \times Q} \ell_i(u, v) = |P||Q| \bar{\ell}_i = T \bar{\ell}_i. \tag{6}$$

Multiplying the updates yields Eq. 3, independent of pair order. Once the normalized matrices are available, fitting the weights takes $O(N(|P| + |Q|))$ operations and test aggregation takes $O(N n_{\mathrm{test}})$. These costs exclude fitting and running the detector agents.

For bounded losses in $[0, 1]$, the standard Hedge calculation gives

$$\sum_{t=1}^{T} \langle \alpha^{(t)}, \ell^{(t)} \rangle - \min_i \sum_{t=1}^{T} \ell_i^{(t)} \leq \frac{\log N}{\eta} + \frac{\eta T}{8}. \tag{7}$$

Indeed, the log normalization factor at step t is at most $-\eta \langle \alpha^{(t)}, \ell^{(t)} \rangle + \eta^2/8$ by Hoeffding's lemma, while the cumulative log normalization is at least $-\log N - \eta \sum_t \ell_i^{(t)}$ for any fixed agent i. Combining these inequalities proves the bound.

Writing $\Delta_i^{(t)} = 1 - 2\ell_i^{(t)}$ and substituting the learning rate in Eq. 3 gives

$$\frac{1}{T} \sum_{t=1}^{T} \langle \alpha^{(t)}, \Delta^{(t)} \rangle \geq \max_i \frac{1}{T} \sum_{t=1}^{T} \Delta_i^{(t)} - \sqrt{\frac{2 \log N}{T}}. \tag{8}$$

This deterministic bound evaluates the pre-update mixture sequence on the supplied losses. The experiments use its terminal weights on separate test scores, so the bound is not a generalization guarantee or a test-AUC bound. Reused calibration pairs are not independent annotations. In particular, Eq. 3 is equivalently $\alpha_i \propto \exp(\sqrt{2T \log N}\, \widehat{\Delta}_i)$: replicating comparison rows can sharpen the weights without adding label information.

B.3 Why a Disagreement Penalty Can Reduce Specialist Influence

Consider $\alpha_i' \propto \alpha_i \exp(-\eta \ell_i)$ with $\eta > 0$. If $0 < \alpha_A < 1$ and $\ell_A > \ell_i$ for every $i \neq A$, then

$$\alpha_A' = \frac{\alpha_A e^{-\eta \ell_A}}{\sum_i \alpha_i e^{-\eta \ell_i}} < \alpha_A, \tag{9}$$

because the denominator is strictly greater than $e^{-\eta \ell_A}$. The calculation does not depend on the observation's label.

For example, let $\ell_i = g(c_i, |S_{ti} - \sum_j \alpha_j S_{tj}|)$, where g is nondecreasing in confidence and strictly increasing in deviation. If agent A deviates strictly more than every other agent and is at least as confident, its loss is strictly largest and its influence decreases. The conclusion applies to this specified loss family; other aggregation rules require their own analysis. A confident, correct specialist can satisfy these conditions when other detectors share a blind spot.

B.4 Covariance Identities and Their Scope

Theorem 1 allows unrestricted labels. Structural assumptions can instead make relative quality identifiable. Spectral methods for classifier accuracy estimation provide one starting point (Dawid & Skene, 1979; Parisi et al., 2014; Jaffe et al., 2015).

Lemma 1 (Rank separation and AUC)**.** *For empirical midranks divided by n, with both classes present,*

$$\Delta_i = \overline{S_i}^{+} - \overline{S_i}^{-} = \mathrm{AUC}_i - \tfrac{1}{2}. \tag{10}$$

The corresponding population identity holds for the marginal mid-distribution transform $F_i^{\mathrm{mid}}(s) = \mathbb{P}(s_i(X) < s) + \frac{1}{2}\mathbb{P}(s_i(X) = s)$.

The separation $\Delta(f) = \mathbb{E}[f(X) \mid Y = 1] - \mathbb{E}[f(X) \mid Y = 0]$ is linear in f, so it extends naturally to a weighted score. The AUC identity applies to each rank-transformed detector; a weighted average need not itself be rank-transformed, and its AUC is not the weighted average of detector AUCs.

Assumption 1 (Conditional independence)**.** *The population rank-transformed scores are mutually independent given Y.*

Theorem 2 (Population covariance identity)**.** *Under Assumption 1, with $p = \mathbb{P}(Y = 1) \in (0, 1)$, for $i \neq j$,*

$$C_{ij} = \mathrm{Cov}(S_i, S_j) = p(1-p)\Delta_i \Delta_j. \tag{11}$$

If at least three Δ_i are nonzero, these off-diagonal entries identify $\sqrt{p(1-p)}\boldsymbol{\Delta}$ up to global sign. The sign is fixed if $\sum_i \Delta_i > 0$ is assumed.

The identity follows from total covariance. Recovering the latent rank-one matrix requires its missing diagonal to be completed; simply zeroing the observed diagonal and taking an eigenvector is a heuristic, not the exact population recovery theorem. Our experiments evaluate that common heuristic: the eigenvector with largest absolute eigenvalue is oriented to have positive sum, clipped at zero, and normalized into weights.

Proposition 1 (Competing label and nuisance components)**.** *Suppose a bounded-score factor model has $S_i = a_i Y + b_i Z + \varepsilon_i$, with Y independent of Z, $\mathrm{Var}(Z) = \sigma_Z^2$, and mutually independent zero-mean residuals independent of (Y, Z). Its cross-covariances are*

$$C_{ij} = p(1-p)a_i a_j + \sigma_Z^2 b_i b_j \quad (i \neq j). \tag{12}$$

The two full rank-one components have spectral norms $A = p(1-p)\|a\|^2$ and $B = \sigma_Z^2 \|b\|^2$. If $a \perp b$, the leading direction of their sum is the label direction when $A > B$, and the nuisance direction when $B > A$. For general loadings it depends on their angle as well as these magnitudes.

For fixed loadings, the label component is $O(p)$ as $p \to 0$, while a shared nuisance can remain $O(1)$. In the orthogonal case the crossover is $\sigma_Z \|b\|/\|a\| = \sqrt{p(1-p)}$, about 0.032 at prevalence 0.001. This is a property of the stipulated factor model, not a universal threshold for empirical zero-diagonal spectral weighting. Its implication is more general: a large leading component can reflect shared behavior that is uninformative about labels. Neither low-rank covariance nor a large eigengap alone validates conditional independence or identifies that component as quality.

B.5 Rank Separation, Including Ties

Proof of Lemma 1. Let $n_+ = k$, $n_- = n - k$, and let $R_+ = \sum_{t:y_t=1} r_{ti}$. The rank-sum identity and the total sum of ranks give

$$\begin{aligned}\Delta_i &= \frac{R_+}{nk} - \frac{n(n+1)/2 - R_+}{n(n-k)} \\ &= \frac{R_+}{k(n-k)} - \frac{n+1}{2(n-k)} = \mathrm{AUC}_i - \frac{1}{2}.\end{aligned}$$

Average ranks make the calculation valid with ties. For the population version, let $h(a,b) = \mathbf{1}\{a > b\} + \frac{1}{2}\mathbf{1}\{a = b\}$. Exchangeability gives $\mathbb{E}h(s_i(X^+), s_i(X^{+\prime})) = 1/2$ and similarly for negatives. Conditioning the reference draw in the marginal mid-distribution transform on its class gives $\mathbb{E}[S_i \mid Y = 1] = p/2 + (1-p)\,\mathrm{AUC}_i$ and $\mathbb{E}[S_i \mid Y = 0] = p(1 - \mathrm{AUC}_i) + (1-p)/2$. Subtracting proves the identity. □

For an arbitrary weighted score, separation remains linear: $\Delta(S\alpha) = \sum_i \alpha_i \Delta_i$. Re-ranking the weighted score preserves its AUC but changes its separation. Consequently, a bound on this linear separation cannot simply be relabelled as a bound on ensemble AUC.

B.6 Covariance Recovery and the Diagonal

Proof of Theorem 2. By total covariance and conditional independence,

$$C_{ij} = \mathbb{E}[\mathrm{Cov}(S_i, S_j \mid Y)] + \mathrm{Cov}(\mathbb{E}[S_i \mid Y], \mathbb{E}[S_j \mid Y]) = p(1-p)\Delta_i\Delta_j.$$

Set $b_i = \sqrt{p(1-p)}\Delta_i$. For any three distinct indices with nonzero loadings, $b_i^2 = C_{ij}C_{ik}/C_{jk}$. Choosing the sign of one nonzero loading then determines every other loading from its cross-product with that entry; zero loadings have zero cross-products with all nonzero ones. The only remaining ambiguity is the simultaneous sign change of all loadings. The stated positive-sum assumption resolves it. □

With only two nonzero loadings, their product alone does not identify the individual magnitudes. Also, the matrix obtained by setting the diagonal to zero is $bb^\top - \mathrm{diag}(b_i^2)$, which generally has a different leading eigenvector from $bb^\top$. The experimental spectral heuristic and population completion argument are therefore distinct. A finite-sample analysis would need to control covariance estimation, diagonal reconstruction, and the relevant eigengap; no such guarantee for the implemented frozen coordinator is claimed here.

Proof of Proposition 1. Independence eliminates cross-terms, yielding $\mathrm{Cov}(S_i, S_j) = a_i a_j \,\mathrm{Var}(Y) + b_i b_j \,\mathrm{Var}(Z)$ for $i \neq j$. The two full outer-product matrices have norms A and B. If $a \perp b$, their normalized loadings are eigenvectors with eigenvalues A and B. The leading direction changes at $A = B$ (where it is not unique). For nonorthogonal loadings their sum acts on $\mathrm{span}\{a, b\}$ and the eigenvector also depends on the angle. For example, if a and b are parallel, both components have the same direction regardless of A/B. □

The model is an assumption on the transformed scores. An arbitrary additive raw-score model need not remain additive after rank normalization. The $O(p)$ comparison holds for fixed loadings and nuisance variance; if these change with prevalence, that dependence must also be accounted for.

B.7 Comparison Against Unlabelled Rows

In oracle mode, hidden anomalies in Q are not removed. If a confirmed positive is a representative positive draw and an independent comparison row follows the target mixture with prevalence p, then

$$\mathbb{E}[S_i \mid Y = 1] - \mathbb{E}[S_i] = (1-p)\big(\mathbb{E}[S_i \mid Y = 1] - \mathbb{E}[S_i \mid Y = 0]\big). \tag{13}$$

The population comparison therefore attenuates class separation without changing its sign under these assumptions. Selected confirmed positives, in-sample calibration scores, and finite candidate pools need not satisfy them. Corrupting the declared positive set introduces a further bias that this identity does not address.

C Coordinator Specifications

The fixed-pool comparisons change the numerical coordination rule while retaining each split's detector-score matrices. Nonnegative weight vectors are normalized by their sum; a zero vector falls back to uniform weights. Logistic and boosted stacking produce scores directly and need not form a convex combination of detector scores.

C.1 Label-Free Rules

Uniform, median, and maximum aggregation act row by row. The trimmed mean drops the smallest and largest score before averaging. AOM partitions a random agent permutation into pairs, takes a maximum within each pair, and averages the pair maxima; MOA takes the maximum of the pair averages. Each averages the outputs of five random partitions. If a pool has an odd number of valid agents, the unmatched entry is omitted in that partition. Consensus pruning retains agents whose Pearson correlation with the uniform score is at least the median correlation. Greedy aggregation starts with the most consensus-correlated agent, then adds agents in descending correlation order only when correlation of the enlarged average with the uniform-score target increases. These are specified consensus-based implementations, not complete reproductions of every method sharing those ideas.

Agreement weights use the mean off-diagonal correlation of each rank-normalized score column with its peers, subtract the minimum across agents, add 10^{-9}, and normalize. The primary spectral heuristic zeros the diagonal of the score covariance, takes the eigenvector associated with the largest absolute eigenvalue, orients it to have positive sum, clips negative entries to zero, and normalizes. The descriptive density membership calculation in Table 7 instead uses the largest-eigenvalue vector of the full covariance. These archived spectral calculations differ in their diagonal treatment; the primary coordinator comparisons use the zero-diagonal version.

Consensus-loss EG begins uniformly and visits up to 3000 test-score rows sampled without replacement. On a row t, it uses confidence $c_{ti} = 2|S_{ti} - 1/2|$, loss $\ell_{ti} = \mathrm{clip}_{[0,1]}[c_{ti}(S_{ti} - \sum_j \alpha_j S_{tj})^2]$, and learning rate one. These methods may inspect unlabelled test-score structure, whereas label-driven EG uses calibration scores to learn its weights. Their information access differs in this respect as well as in label access.

C.2 Label-Driven Rules

Estimated-best selection maximizes empirical AUC against all other eligible candidates assigned normal pseudo-labels. AUC reweighting normalizes $(\widehat{\mathrm{AUC}}_i - 1/2)_+$. EG uses the exact mean-separation update in Eq. 3. Mirrored EG augments the pool with $1 - S_i$ for each score column and applies the same update to $2N$ experts; it can reverse a detector's ordering.

The Thompson-style reference counts strict ordering successes over at most 2000 anomaly–comparison pairs sampled without replacement. Tied pairs are not successes in this implementation. With u_i successes in T' pairs, it samples 200 vectors from independent $\mathrm{Beta}(1 + u_i, 1 + T' - u_i)$ distributions, normalizes the positive excess above $1/2$ for each vector, and averages the resulting weights. The UCB-style reference normalizes

$$\left(\frac{u_i}{T'} + \sqrt{\frac{2 \log \max(T', 2)}{T'}} - \frac{1}{2} \right)_+ . \tag{14}$$

These are offline weighting references, not interactive bandit policies. Pair reuse does not increase the number of independent labels.

Logistic stacking uses rank-score columns as features, balanced class weights, regularization parameter $C = 1$, and at most 500 iterations. Boosted stacking uses histogram gradient boosting with 100 iterations, maximum depth three, learning rate 0.1, and balanced class weights. Both fit to P and the same Q used by EG. Table 2 reports their matched-budget results.

C.3 Selection and Pool-Rebuilding Controls

Model centrality and HITS select a detector using unlabelled score structure (Ma et al., 2023). Excess-mass and mass-volume selection require additional scoring operations (Goix, 2016); these

Table 6: Additional label-free controls on real anomalies. Values are mean ROC-AUC, gain over the appropriate uniform reference, and dataset wins/losses. Superscript † marks the nine-agent reference (0.759); ‡ marks MetaOD's externally meta-trained candidate library with benchmark overlap. Other rows use uniform-10 (0.763). Stars denote paired dataset-level Wilcoxon tests against the reference: $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$.

Rule	AUC	Gain	Wins/losses	Reference
Trimmed mean	0.766	$+0.003^{*}$	28/15	
Maximum	0.764	$+0.002$	22/23	Aggarwal & Sathe (2017)
Maximum of averages (MOA)	0.770	$+0.007$	23/22	Aggarwal & Sathe (2017)
Greedy ensemble	0.765	$+0.002$	25/19	Schubert et al. (2012)
Model centrality (MC)	0.758	-0.005	20/25	Ma et al. (2023)
HITS	0.756	-0.006	19/26	Ma et al. (2023)
Excess-mass, nine agents†	0.751	-0.008	27/18	Goix (2016)
Mass-volume, nine agents†	0.742	-0.017	23/22	Goix (2016)
MetaOD, 302 candidates‡	0.671	-0.092^{**}	14/31	Zhao et al. (2021)
MetaOD, top-3 average‡	0.691	-0.071^{*}	15/30	Zhao et al. (2021)
LSCP, nine agents†	0.749	-0.009^{*}	16/28	Zhao et al. (2019)
Feature bagging (LOF)†	0.657	-0.102^{**}	17/28	Lazarevic & Kumar (2005)

controls refit the nine classical agents locally because the density model could not be re-scored in that CPU evaluation. LSCP and feature bagging rebuild or locally select from a pool (Zhao et al., 2019; Lazarevic & Kumar, 2005). Their reference is the corresponding nine-agent uniform score, whose mean AUC is 0.759. MetaOD uses its released meta-learner and 302 candidate configurations (Zhao et al., 2021); its external meta-training corpus overlaps ADBench. It is therefore a differently scoped meta-learned baseline rather than a disjoint-meta-training comparison restricted to the TabMATE pool.

D Individual Expertise and Budget-Dependent Coordination

Table 7 gives the numeric agent-level results underlying Figure 3, including the change from adding the density specialist to the uniform nine-agent pool. The retrospective best local agent selects among KNN, LOF, and CBLOF using test AUC for analysis only. This reference is not available to the coordinator. Tables 8 and 9 show how performance and influence vary with anomaly type and feedback budget. Their budget-specific means use datasets eligible at each budget.

Query budgets count requested annotations, whereas oracle budgets count supplied anomalies. The primary query-30 experiment has 450 draws and finds 3.6 anomalies on average; 113 draws find none and return uniform weights. The external and expanded-pool comparison uses two draws per seed rather than five, giving 180 query-30 draws. Its positive-discovery aggregates retain 129 draws from 41 datasets. The all-query policy in Table 3 restores the other 51 draws with a uniform-10 fallback. For TabICL and EG-11, this is a composite policy that uses the fallback when a label-conditioned model cannot be fitted.

A historical query experiment used a nine-agent pool and TabPFN v1 on 33 datasets, two seeds, and an evaluation half capped at 2000 rows. Table 10 compares score-based row selection with random queries. Its settings and model generation differ from the primary TabPFN-density/TabICL study, so its results are not pooled into the main aggregates. Inputs are capped at 8000 rows, at most 64 principal components, and a complete TabPFN context of at most 1000 rows (revealed rows plus unlabelled fill). Pool reweighting uses AUC estimated from the revealed labels and at most 2000 unlabelled training rows treated as normal. The training rows supply both classical-agent fitting and calibration scores. Query disagreement is the variance across normalized agent scores; the combined policy ranks disagreement within the consensus top-10% set. Each method falls back to the uniform pool if the revealed set cannot support its required positive-class fit.

Selecting high-consensus rows often discovers more anomalies than random selection, but these labels need not be the most informative for choosing among agents. The results characterize the implemented heuristics and leave open whether other score-based query policies can improve coordination.

Table 7: Individual agents and the effect of density scoring. Mean AUC on 45 datasets and two seeds, including four generated anomaly types. Density denotes TabPFN v2. The local-best reference selects the best of KNN, LOF, and CBLOF per dataset. Ranks, win fractions, and pool gains use dataset summaries. Stars denote paired Wilcoxon tests ($^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$).

	Real	Local	Cluster	Global	Depend.
HBOS	0.724	0.837	0.961	0.988	0.595
COPOD	0.734	0.857	0.963	0.989	0.612
ECOD	0.732	0.882	0.940	0.990	0.601
KNN	0.764	0.897	0.882	0.996	0.864
LOF	0.657	0.923	0.543	0.950	0.891
CBLOF	0.736	0.891	0.855	0.997	0.856
PCA	0.725	0.868	0.978	0.991	0.645
MCD	0.761	0.802	0.904	0.985	0.866
IForest	0.749	0.886	0.969	0.995	0.793
TabPFN density	0.716	0.895	0.806	0.996	0.953
Density rank (median)	8	2	9	1	1
Density is best	4%	38%	9%	56%	80%
Density − local best	-0.075^{***}	-0.034^{**}	-0.103^{***}	-0.002^{**}	$+0.044^{***}$
Wins vs. local best	20%	40%	7%	11%	82%
uniform-10 − uniform-9	$+0.004^{**}$	$+0.004^{**}$	-0.007^{***}	$+0.001^{***}$	$+0.029^{***}$
Full-covariance spectral weight	0.081	0.093	0.091	0.095	0.077
Density agreement rank	9	8	8	8	9

Table 8: Coordination gains across feedback budgets. Gains over uniform-10 for the primary pool, with two seeds and five draws per dataset. Oracle-k supplies k anomalies; query-B labels B random rows and includes zero-positive fallback. Best-agent gap uses retrospective selection. Gains are reported without significance annotations.

Type	Budget	Spectral	AUC weights	EG	Best-agent gap
Real	oracle-3	+0.000	+0.031	+0.063	+0.073
	oracle-10	−0.000	+0.042	+0.148	+0.077
	query-30	+0.000	+0.026	+0.060	+0.073
	query-100	−0.000	+0.034	+0.095	+0.069
Local	oracle-3	−0.002	+0.007	+0.014	+0.040
	oracle-10	−0.003	+0.009	+0.029	+0.039
	query-30	−0.002	+0.006	+0.013	+0.040
	query-100	−0.002	+0.008	+0.022	+0.041
Cluster	oracle-3	+0.010	+0.036	+0.047	+0.059
	oracle-10	+0.012	+0.045	+0.066	+0.068
	query-30	+0.010	+0.033	+0.046	+0.059
	query-100	+0.009	+0.038	+0.055	+0.059
Global	oracle-3	+0.000	+0.000	−0.003	+0.002
	oracle-10	+0.000	+0.000	+0.000	+0.001
	query-30	+0.000	+0.000	+0.000	+0.002
	query-100	+0.000	+0.000	+0.001	+0.001
Depend.	oracle-3	−0.022	+0.074	+0.109	+0.131
	oracle-10	−0.023	+0.078	+0.121	+0.129
	query-30	−0.022	+0.057	+0.088	+0.131
	query-100	−0.023	+0.069	+0.111	+0.127

Table 9: Detector influence across feedback budgets. Runs match Table 8. Left: weight on TabPFN density. Right: Spearman correlation between weights and agent AUC. Uniform weight is 0.1 in a complete ten-agent pool.

		Density weight		Weight–AUC ρ	
Type	Budget	Spectral	EG	Spectral	EG
Real	oracle-3	0.081	0.085	+0.21	+0.54
	oracle-10	0.079	0.112	+0.20	+0.72
	query-30	0.081	0.112	+0.21	+0.53
	query-100	0.081	0.091	+0.21	+0.66
Local	oracle-3	0.093	0.136	+0.03	+0.48
	oracle-10	0.092	0.164	−0.03	+0.67
	query-30	0.093	0.126	+0.03	+0.45
	query-100	0.093	0.117	+0.02	+0.62
Cluster	oracle-3	0.091	0.015	+0.39	+0.69
	oracle-10	0.089	0.001	+0.41	+0.82
	query-30	0.091	0.049	+0.39	+0.62
	query-100	0.092	0.024	+0.37	+0.74
Global	oracle-3	0.095	0.074	+0.21	+0.20
	oracle-10	0.094	0.075	+0.24	+0.30
	query-30	0.095	0.109	+0.21	+0.15
	query-100	0.095	0.105	+0.24	+0.24
Depend.	oracle-3	0.077	0.349	−0.33	+0.80
	oracle-10	0.076	0.477	−0.34	+0.90
	query-30	0.077	0.310	−0.33	+0.81
	query-100	0.078	0.450	−0.33	+0.88

Table 10: Historical query-selection comparison: nine classical agents and TabPFN v1, 33 datasets and two seeds (988 eligible runs across policies and budgets). "Found" counts discovered anomalies; the three score columns are ROC-AUC gains over the uniform nine-agent pool. The oracle row supplies B confirmed anomalies and does not represent B annotation queries.

B	Query policy	Found	Pool	FM	FM + scores
3	oracle: B anomalies handed over	3.0	+0.0375	+0.0953	+0.0919
	random queries	0.4	+0.0077	+0.0074	+0.0013
	top-B by pool consensus	1.5	+0.0002	+0.0006	−0.0023
	top-B by pool disagreement	0.4	+0.0118	+0.0085	+0.0041
	Disagreement in consensus top-10%	1.3	+0.0040	+0.0060	+0.0058
10	oracle: B anomalies handed over	10.0	+0.0475	+0.1412	+0.1447
	random queries	1.1	+0.0159	+0.0262	+0.0222
	top-B by pool consensus	4.5	+0.0005	+0.0083	+0.0043
	top-B by pool disagreement	1.9	+0.0185	+0.0277	+0.0147
	Disagreement in consensus top-10%	4.1	+0.0060	+0.0172	+0.0120
30	oracle: B anomalies handed over	30.0	+0.0455	+0.1683	+0.1688
	random queries	4.0	+0.0317	+0.0895	+0.0933
	top-B by pool consensus	12.1	+0.0014	+0.0252	+0.0187
	top-B by pool disagreement	5.9	+0.0304	+0.0569	+0.0382
	Disagreement in consensus top-10%	11.2	+0.0088	+0.0419	+0.0352

Table 11: Spectral-weight diagnostics. Nine classical agents on 38 datasets and five seeds. Left: median covariance diagnostics. Right: median Spearman correlation with agent AUC and the fraction of runs with correlation above 0.5.

Covariance structure (median)		*Statistic*	Median ρ	$P(\rho > 0.5)$
λ_1/λ_2	5.68	Leading eigenvector v	$+0.283$	25%
Leading energy share	0.86	Mean peer correlation	$+0.300$	25%
$\rho(v,$ agreement$)$	0.983	Tail agreement	$+0.195$	27%
Fraction of runs > 0.9	96%	Consensus tail overlap	$+0.289$	27%
$\rho(v,$ AUC$)$	0.283	Raw-score skewness	$+0.200$	16%
		Raw-score kurtosis	$+0.158$	14%
		Raw-score variance	-0.017	9%

Table 12: Classical-agent coordination. Gains over uniform weighting on 38 datasets. Paired Wilcoxon tests use dataset means over five seeds; wins/losses require $|\Delta| > 10^{-4}$.

Rule	AUC gain	p	Dataset wins/losses
Spectral heuristic	$+0.0005$	0.468	21/16
Agreement weighting	-0.0029	0.852	22/16
Anti-agreement weighting	-0.0268	0.0274	15/23
Consensus-loss EG	-0.0144	0.261	19/19
Best single detector (oracle)	$+0.0752$	7.28e-12	38/0

E Spectral Diagnostics and the Stronger-Minority Experiment

The nine-agent diagnostic uses 38 datasets and five seeds, separately from the primary ten-agent experiment. Table 11 reports covariance and per-agent quality diagnostics; Table 12 reports detection gains. Large spectral concentration coexists with weak AUC alignment: the leading vector correlates more strongly with agreement than with quality. These descriptive results characterize this detector pool and do not directly test conditional independence.

The stronger-minority experiment adds five supervised specialists to nine unsupervised agents on 31 datasets and five seeds. The supervised members are XGBoost (150 trees, depth three, learning rate 0.1), LightGBM (150 trees, depth four, minimum child size five), random forest (150 trees), logistic regression (2000 iterations), and a multilayer perceptron with hidden widths 64 and 32 (300 iterations). Random state zero is supplied to the tree models and MLP. Unsupervised scores are fitted once per split and reused across training budgets. The data cap is 8000 rows and the feature cap is 64 principal components; standardization precedes the stratified 50/50 split. Eligibility requires at least 70 anomalies and 400 rows before splitting, at least 30 training anomalies, and at least 15 test anomalies.

Finite-budget training samples $k \in \{5, 10, 25\}$ anomalies and up to $20k$ normals from the training half; full supervision uses the entire training half. These extra labels train the specialist group and are not coordinator calibration labels. "Underweighted" means that the group with higher mean test AUC receives less than its count-proportional spectral weight. Fractions and correlations summarize runs, while the reported paired tests average seeds within datasets. Table 13 provides the full results underlying Figure 3.

Exploratory variants include tail-restricted covariance, shrinkage, anti-agreement, and third-order statistics. None provided a consistently effective correction across the recorded settings. This empirical finding does not establish a universal impossibility result. The algorithms were developed while inspecting these benchmarks, so the comparison is exploratory rather than preregistered on a fresh suite.

F Foundation-Model and Label-Noise Controls

F.1 External Models and Computation

External models receive the same revealed labels as the corresponding coordinator draw. TabPFN v2 classification, TabICL, and TabDPT use in-context examples: revealed positives, revealed negatives in

Table 13: The stronger supervised minority across training budgets. Top: group AUC and agreement. Bottom: agreement–AUC correlation, spectral AUC gain over the uniform mixed pool, paired dataset-level significance, and underweighting frequency. The 38-dataset unsupervised-only reference is separately reported in Table 12.

k	Sup. AUC	Unsup. AUC	Sup. agreement	Unsup. agreement
5	0.849	0.728	0.303	0.482
10	0.882	0.728	0.331	0.492
25	0.907	0.728	0.311	0.478
full	0.941	0.728	0.311	0.469
k	Correlation	Spectral gain	p	Underweighted
5	-0.068	-0.0359	$1.8e-08$	79%
10	-0.138	-0.0357	$6.5e-08$	88%
25	-0.218	-0.0467	$4e-08$	91%
full	-0.244	-0.0551	$2.9e-07$	92%

query mode, and up to 1000 additional unlabelled training rows assigned normal pseudo-labels. The TabPFN classifier uses four estimators. TabICL uses its classifier constructor with the split random seed; TabDPT uses its classifier API and passes the split seed to prediction. The comparison matches revealed labels, while pretraining resources, computation, and numerical interfaces differ. The direct-prediction and expanded-pool blocks in Table 4 use separate matched runs; EG-10 references in Tables 14–16 belong to the expanded-pool study.

DevNet, PReNet, DeepSAD, FEAWAD, REPEN, FT-Transformer, and ResNet use the ADBench implementations. Their training target marks only revealed positives as anomalous and treats the remaining training rows as normal. DeepSAD's training and pretraining batch sizes are capped at $\min(128, n_{\text{train}})$ to avoid an empty loader on small datasets. XGBoost on raw features uses 150 trees, depth three, learning rate 0.1, histogram tree construction, and positive-class weight equal to the normal-to-positive training-count ratio (lower bounded by one). The XGBOD-style control fits the same classifier to calibration rows with raw features concatenated with the fixed detector-score columns; it does not train on every row of the full training half.

TabM uses 32 ensemble members, AdamW with learning rate 2×10^{-3} and weight decay 3×10^{-4}, 50 epochs, and batch size 256. Binary cross-entropy is weighted by the normal-to-positive count ratio, and member logits are averaged at prediction. Apart from these explicit adaptations, the baselines use their implementation defaults. Small-label results therefore characterize these fitted configurations rather than an exhaustive tuning budget for each model.

Table 4 reports external-model results and measured seconds per run on an H200. External timings include model construction, fitting, prediction, and AUC computation. The fixed pool takes 84.3 seconds on average, including 82.8 seconds for density scoring; EG calibration adds less than 0.1 second once scores are available. These timings cover different operations and should not be read as inference-only latency.

F.2 Calibrating a Label-Conditioned Extra Agent

Adding TabICL as an eleventh detector requires a calibration score for each row without showing that row to the model as an in-context example. The full-context model produces test scores, while held-out context variants produce calibration scores. Each revealed positive is scored after leaving it out of context. Revealed negatives are held out in groups of at most five; unlabelled context rows that also belong to the calibration pool are held out in two folds. Calibration rows outside the context use the full-context model. The predictions form one rank-normalized score column. Thus, each calibration row is scored without its own context entry, but the column combines differently conditioned models.

At $k = 1$, leaving out the only positive removes the positive class, making its held-out score undefined. The implementation fills that entry with the median of the remaining calibration scores and flags the case. This fallback can lower the extra agent's estimated quality even when its full-context test model is informative. With more positives, the held-out models retain positive examples, but calibration and test conditioning still differ. Team–TabICL comparisons therefore depend on the label budget and on

Table 14: Adding TabICL as an eleventh agent. Gains over uniform-10 with two seeds and the first two draws. TabICL receives the coordinator's revealed labels. Calibration rows are held out of context; at $k = 1$, the undefined held-out positive score is replaced by the median of the remaining calibration scores. There are 45 datasets at $k = 1, 3$ and 38 at $k = 10$. Query-30 conditions on finding an anomaly (129/180 draws; 41 datasets); Table 3 includes all draws. Gains are reported without significance annotations.

Rule	k=1	k=3	k=10	B=30
Uniform-10	+0.000	+0.000	+0.000	+0.000
EG-10	+0.021	+0.047	+0.064	+0.048
AUC weights, 10	+0.020	+0.032	+0.041	+0.032
Logistic stacking, 10	+0.009	+0.060	+0.102	+0.065
Best of 10 (oracle)	+0.073	+0.073	+0.077	+0.076
TabICL alone	+0.053	+0.101	+0.161	+0.113
Uniform-11	+0.011	+0.016	+0.024	+0.018
Select estimated best, 11 agents	+0.014	+0.062	+0.129	+0.059
EG-11	+0.061	+0.109	+0.129	+0.080
AUC weights, 11	+0.021	+0.049	+0.089	+0.055
Logistic stacking, 11	−0.016	+0.031	+0.101	+0.081
Best of 11 (oracle)	+0.095	+0.119	+0.165	+0.123
TabICL weight (EG / AUC)	0.06 / 0.02	0.21 / 0.10	0.49 / 0.18	0.22 / 0.10
TabICL rank (mean)	3.5	2.2	1.3	2.0

how the extra agent is calibrated; they do not isolate the value of adding a strong model. Table 14 reports the expanded-pool alternatives and the extra member's weight and rank. EG-11 gains 0.061, 0.109, and 0.129 at one, three, and ten confirmed anomalies, compared with 0.053, 0.101, and 0.161 for TabICL alone. The team leads at one and three anomalies, while the individual model leads at ten.

F.3 Score Augmentation and Label Noise

The augmentation control gives TabICL raw features, detector scores, or their concatenation, using the same revealed rows and unlabelled context sampled from the calibration candidates. Table 15 provides supplementary comparisons across the one-, three-, and ten-label oracle settings and the query setting. The paired raw-versus-concatenated comparison uses identical context; no candidate model is selected using test labels.

The label-noise control replaces one or three supplied anomalies with randomly sampled normal candidate rows, identically across methods. TabICL is reconditioned and its extra-agent calibration is repeated using the corrupted declared labels. This tests false-positive contamination of an oracle-provided anomaly set. Table 16 reports the coordinator comparisons, including three wrong labels. The available records lack per-draw TabICL prediction archives, so the controls support reaggregation of recorded metrics but not an independent audit of each model's raw predictions.

A four-dataset FoMo-0D reproduction check did not reproduce the expected accuracy, so it is not included as a competitive full-benchmark baseline. OutFormer is not evaluated in these experiments. These omissions leave zero-shot foundation-model coverage incomplete; no general comparison against all current foundation models is claimed.

G Language-Model Interfaces and Detailed Results

The detector agents supply numerical anomaly scores. This supplementary study evaluates LLMs in two optional roles: a coordinator assigns detector weights from summarized evidence, and a reporter interprets detector reports and weights to identify anomaly structure. The following protocol and results describe each role separately.

G.1 Coordinator Inputs and Response Processing

Each LLM coordinator prompt contains a data card and one report per numerical detector. The card records row and feature counts, fractions of binary and low-cardinality features, mean absolute

Table 15: Detector scores as TabICL features. Inputs are raw features x, agent scores, or their concatenation. Calibration-pool context uses the same revealed rows and up to 1000 unlabelled calibration rows. Gains are over uniform-10 with two seeds and two draws; 45 datasets at $k = 1, 3$, 38 at $k = 10$. Query-30 conditions on finding an anomaly (129/180 draws; 41 datasets). Paired feature comparisons use the same context. Gains are reported without significance annotations.

Method	k=1	k=3	k=10	B=30
EG-10	+0.021	+0.047	+0.064	+0.048
EG-11	+0.061	+0.109	+0.129	+0.080
TabICL: x, training context	+0.053	+0.101	+0.161	+0.113
TabICL: x, calibration context	+0.052	+0.102	+0.163	+0.113
TabICL: scores only	−0.073	+0.028	+0.107	+0.035
TabICL: $[x, \text{scores}]$	−0.009	+0.073	+0.156	+0.089
Best of 11 (oracle)	+0.095	+0.119	+0.165	+0.123
$[x, \text{scores}] - x$, same context	−0.061	−0.029	−0.007	−0.024
$[x, \text{scores}]$ − EG-11	−0.070	−0.036	+0.027	+0.009

Table 16: False-positive contamination of the revealed anomaly set. One or three supplied anomalies are replaced by normal rows, identically across methods. Gains are over uniform-10; 45 datasets at $k = 3$, 38 at $k = 10$, two seeds and two draws. The final row gives EG weight on TabICL and its mean cross-fitted AUC estimate. Clean columns match Table 14. Gains are reported without significance annotations.

Rule	k=3	3, 1 wrong	k=10	10, 1 wrong	10, 3 wrong
EG-10	+0.047	+0.017	+0.064	+0.057	+0.042
AUC weights, 10	+0.032	+0.024	+0.041	+0.040	+0.034
Logistic stacking, 10	+0.060	−0.002	+0.102	+0.085	+0.049
TabICL alone	+0.101	+0.052	+0.161	+0.145	+0.116
EG-11	+0.109	+0.020	+0.129	+0.105	+0.062
AUC weights, 11	+0.049	+0.032	+0.089	+0.081	+0.058
Logistic stacking, 11	+0.031	−0.067	+0.101	+0.064	−0.009
EG-11 − TabICL alone	+0.008	−0.032	−0.032	−0.040	−0.054
TabICL weight / $\widehat{\text{AUC}}$	0.21 / 0.69	0.02 / 0.41	0.49 / 0.77	0.27 / 0.69	0.14 / 0.55

skewness, mean excess kurtosis, mean and maximum absolute feature correlation, and the number of principal components explaining 90% of variance. It omits the anomaly rate. Detector reports describe the statistical family and give mean peer rank correlation, individual pairwise correlations, overlap with the uniform-score top-5% set, and a spectral weight. Mean correlations, overlap fractions, and spectral weights are rounded to three decimals; individual correlations are rounded to two.

The labelled coordinator receives the first oracle-3 draw and eight unlabelled examples sampled from calibration rows outside those three positives. Each agent's position for each shown row is rendered as a percentile rounded to one decimal percentage point. These eight rows are not labelled normal. EG uses the same three positive indices and the full-precision calibration matrix, including all its comparison rows. Thus, the experiment compares usable interfaces and does not isolate reasoning with identical numerical input.

The recorded coordinator instruction is:

> You are the coordinator of a pool of anomaly-detection agents for one tabular dataset. Each agent has already scored every row; scores are rank-normalised to (0,1], higher = more anomalous. Your job is to decide how much to trust each agent: output a weight vector over the agents (non-negative, summing to 1). The final anomaly score of a row is the weighted average of the agents' rank scores, and it will be evaluated by ROC-AUC against hidden ground-truth anomaly labels. Use everything you know about when each kind of detector works. Answer with a single JSON object containing "weights" (one number per agent) and "rationale" (at most 120 words). Every agent must appear in "weights". No other text.

Table 17: LLM coordinator results. First oracle-3 draw on 45 datasets and two seeds. Gains and stars use paired dataset means; wins/losses count datasets. Weight ρ measures alignment with agent AUC. Labelled LLMs receive rounded reports and eight unlabelled examples; EG uses full-precision calibration scores.

Coordinator Backbone	Input	AUC	ΔAUC	Wins/losses	Weight ρ	LLM − EG
Uniform average		0.763	–	–	–	–
Spectral (label-free)		0.763	$+0.000$	–	–	–
EG, 3 labels		0.812	$+0.050$	–	–	–
Best single (oracle)		0.836	$+0.073$	–	–	–
Claude Sonnet 4.6	Reports	0.768	$+0.005$	27/17	$+0.29$	–
	+ 3 labels	0.781	$+0.018^{***}$	33/11	$+0.46$	-0.032^{***}
GPT-4o	Reports	0.763	$+0.000$	22/22	$+0.24$	–
	+ 3 labels	0.770	$+0.007^{**}$	30/14	$+0.34$	-0.042^{***}
Qwen3-32B	Reports	0.767	$+0.004^{*}$	29/16	$+0.23$	–
	+ 3 labels	0.771	$+0.008^{***}$	33/11	$+0.31$	-0.041^{***}

The final two sentences above preserve the recorded schema in prose to avoid an overwide JSON display. Each user message presents the data card, agent reports, optional revealed-row card, and the agent-name list before requesting the JSON response. Parsed weights are clipped to be nonnegative and normalized. A zero vector returns uniform weights.

G.2 Reporter Task and Reference Readout

The reporter receives the data card and detector reports, optionally augmented with spectral weights or EG weights after three confirmed anomalies. Its four-way task is to identify the generated anomaly type on 17 datasets at seed zero. The recorded instruction defines local anomalies as locally sparse observations within clusters, clustered anomalies as small dense groups away from normal clusters, global anomalies as extreme feature values, and dependency anomalies as ordinary marginal values that break feature relationships. It requests one type, a confidence in $[0, 1]$, and an explanation of at most 100 words in a JSON object. Explicit supplied weights are rounded to three decimals. The "reports only" condition already contains the spectral summary within each detector report, so the separate spectral-weight condition makes that same summary explicit rather than adding labels.

The logistic readout uses the EG weight vector as features, $C = 1$, and at most 2000 iterations. It trains on the other 16 datasets for each held-out dataset, using their generated type labels. Its additional supervision differs from the LLM reporter prompt; it is a reference for type information recoverable from the weights, not a label-free matched baseline. Reporter identifiers are blinded, and the earlier unblinded pilot is excluded from the reported results.

Both LLM roles use temperature zero and one response per prompt. The recorded backbones are Claude Sonnet 4.6, GPT-4o (2024-11-20), and Qwen3-32B with thinking disabled. The experiment records contain prompt templates, a blinded reporter mapping, pilot responses, named-backbone metric files, and usage totals, but not the named backbones' raw coordinator or reporter responses. The tables can therefore be checked against recorded metrics; outputs have not been independently reparsed from raw text. Table 17 reports coordinator performance, dataset wins/losses, and weight-quality diagnostics. Table 18 reports accuracy by true anomaly type.

H Dataset Coverage and Statistical Reporting

Table 19 separates the experiment families. The 45-dataset primary pool, 38-dataset classical diagnostic, and 31-dataset mixed-pool study are not merged into a single aggregate. At oracle-10, insufficient eligible positives reduce the primary comparison to 38 datasets. Table 20 identifies the primary datasets and the subsets used for the larger label budget and reporter study.

Unless stated otherwise, results first average seeds and calibration draws within a dataset, then average datasets with equal weight. Paired Wilcoxon signed-rank tests operate on dataset-level differences. Stars encode $p < .05$, $p < .01$, and $p < .001$ without multiplicity correction; comparisons are exploratory. Dataset win/loss columns use the convention stated in each table. The mixed-pool run

Table 18: LLM reporter accuracy on 17 datasets and four generated types at seed zero. The four-class chance reference is 0.25. The logistic weight readout is trained using type labels from other datasets. "Reports only" already includes per-agent spectral summaries.

Backbone	Input	All	Local	Clustered	Global	Dependency
Claude Sonnet 4.6	reports only	0.29	0.06	0.00	1.00	0.12
	+ spectral weights	0.25	0.00	0.00	1.00	0.00
	+ EG weights	0.35	0.24	0.00	0.53	0.65
Supervised weight readout		0.56	0.24	0.82	0.41	0.76
GPT-4o	reports only	0.24	0.06	0.12	0.76	0.00
	+ spectral weights	0.25	0.12	0.06	0.82	0.00
	+ EG weights	0.31	0.29	0.00	0.35	0.59
Qwen3-32B	reports only	0.25	0.82	0.00	0.06	0.12
	+ spectral weights	0.22	0.71	0.00	0.06	0.12
	+ EG weights	0.25	0.35	0.00	0.12	0.53

medians and underweighting fractions are descriptive, while its significance tests still use dataset means. Retrospective best-agent results inspect test labels for diagnosis and are neither deployable selection rules nor upper bounds on all ensembles.

The weight intervals are percentile bootstrap intervals from 5000 resamples of dataset means using random seed 2027. Seeds and draws are not treated as independent datasets. The benchmark caps, finite repeats, transductive preprocessing, in-sample classical calibration, and reliance on ROC-AUC limit transfer to large-scale deployment, temporal drift, and precision-sensitive alerting tasks.

Table 19: Experiment families and their distinct coverage. Counts refer to base datasets; budget eligibility and failed generated cells can reduce the number of individual evaluations.

Family	Coverage	Purpose
Classical diagnostic	38; five seeds	Score covariance and label-free per-agent statistics
Stronger minority	31; five seeds	Nine unsupervised and five supervised agents; four training budgets
Primary detector pool	45; two seeds	90 real-data and 358 generated-type score matrices
Coordinator comparison	Up to 45; five draws per seed	Same detector scores and revealed row indices
External and expanded pool	45 at $k = 1, 3$; 38 at $k = 10$; two draws per seed	Direct prediction, extra TabICL agent, feature augmentation, and noise
Historical query selection	33; two seeds	Nine-agent pool and TabPFN v1; 988 eligible evaluations
LLM coordinator	45; two seeds; one draw	Label-free reports or three revealed anomalies
LLM reporter	17; one seed	Four generated anomaly types; blinded identifiers

Table 20: Primary dataset coverage. n, d, and anomaly prevalence describe the sampled real dataset at seed zero after the documented row and feature caps, before the half split. The last columns mark membership in the oracle-10 and four-type reporter subsets.

Dataset	n	d	Anomaly %	$k = 10$	Reporter
ALOI	10000	27	3.04	Yes	–
annthyroid	7200	6	7.42	Yes	–
backdoor	10000	64	2.44	Yes	–
breastw	683	9	34.99	Yes	Yes
campaign	10000	62	11.27	Yes	–
cardio	1831	21	9.61	Yes	Yes
Cardiotocography	2114	21	22.04	Yes	Yes
celeba	10000	39	2.24	Yes	–

Dataset	n	d	Anomaly %	$k = 10$	Reporter
census	10000	64	6.20	Yes	–
cover	10000	10	0.96	Yes	–
donors	10000	10	5.93	Yes	–
fault	1941	27	34.67	Yes	Yes
fraud	10000	29	0.20	–	–
Hepatitis	80	19	16.25	–	Yes
http	10000	3	0.39	–	–
InternetAds	1966	64	18.72	Yes	Yes
Ionosphere	351	32	35.90	Yes	Yes
Landsat	6435	36	20.71	Yes	–
letter	1600	32	6.25	Yes	Yes
magic.gamma	10000	10	35.16	Yes	–
mammography	10000	6	2.32	Yes	–
mnist	7603	64	9.21	Yes	–
musk	3062	64	3.17	Yes	–
optdigits	5216	62	2.88	Yes	–
PageBlocks	5393	10	9.46	Yes	–
pendigits	6870	16	2.27	Yes	–
Pima	768	8	34.90	Yes	Yes
satellite	6435	36	31.64	Yes	–
satimage-2	5803	36	1.22	Yes	–
shuttle	10000	9	7.15	Yes	–
skin	10000	3	20.75	Yes	–
smtp	10000	3	0.20	–	–
SpamBase	4207	57	39.91	Yes	–
speech	3686	64	1.65	Yes	–
Stamps	340	9	9.12	Yes	Yes
thyroid	3772	6	2.47	Yes	–
vertebral	240	6	12.50	Yes	Yes
vowels	1456	12	3.43	Yes	Yes
Waveform	3443	21	2.90	Yes	–
WBC	223	9	4.48	–	Yes
WDBC	367	30	2.72	–	Yes
Wilt	4819	5	5.33	Yes	–
wine	129	13	7.75	–	Yes
WPBC	198	33	23.74	Yes	Yes
yeast	1484	8	34.16	Yes	Yes