Model-Agnostic Association Rule Learning with Tabular Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

Association Rule Mining (ARM) is a fundamental data mining technique for knowledge discovery in tabular data and is widely used in high-stakes decision-making. Existing approaches suffer from two fundamental limitations: reliance on frequent itemset mining leading to rule explosion (for classical methods), and failure in low-data regimes (for recent neural approaches). We reframe ARM to a conditional probability estimation problem and introduce a model-agnostic rule learning framework that extracts association rules from any conditional probabilistic model, formally decoupling rule extraction from frequent itemset mining. We introduce *TabProbe*, a probing-based algorithm that instantiates this framework, and demonstrate its generality across multiple model classes ranging from gradient-boosted trees to large pretrained foundation models. As a particularly powerful instantiation, we leverage tabular foundation models (TFMs), large models pretrained on diverse tabular data with strong in-context generalization, which can be applied *out-of-the-box* without task-specific training. Evaluation on datasets of varying sizes shows that TFM-learned rules are concise, non-spurious, and highly generalizable, achieving up to 15% higher association strength than state-of-the-art with full data coverage. Furthermore, TFM-learned rules have better predictive performance on unseen data and remain robust in low-data settings. Source code is available at https://anonymous.4open.science/r/tabprobe-D0C0.

1 Introduction

Association Rule Mining (ARM) is an unsupervised knowledge discovery task that aims to find associations between features in a given dataset, such as tabular data, in the form of logical implications (Agrawal et al., 1994). It has been widely used across numerous domains for knowledge discovery (Kaushik et al., 2023; Luna et al., 2019) and high-stakes decision-making (Rudin, 2019) as part of interpretable Machine Learning (ML) models (Angelino et al., 2018). The ARM literature has been dominated by algorithmic methods that suffer from rule explosion and high computational overhead (Moens et al., 2013) without effective search space reduction (Srikant et al., 1997; Baralis et al., 2012; Yin et al., 2022; Fournier-Viger et al., 2012; Nguyen et al., 2018; Zaki & Hsiao, 2002). Such methods typically operate in two phases: mining frequent itemsets, and then generating association rules using the frequent itemsets.

Recent neurosymbolic methods, such as Aerial+ (Karabulut et al., 2025b), utilize autoencoders (Bank et al., 2023) to learn compact representations and extract small rule sets, overcoming rule explosion. However, such methods perform poorly on small datasets (Liu et al., 2017), leading to inaccurate rules that do not reflect true patterns (Karabulut et al., 2025a). This is particularly problematic as small tabular data is common in many domains, including biomedicine, such as rare disease datasets (Banerjee et al., 2023).

To address both rule explosion and low-data scenarios, we propose a novel framing of ARM as a conditional probability estimation problem and introduce a model-agnostic framework that formally decouples association rule extraction from frequent itemset mining, enabling any conditional probabilistic model over tabular data to learn association rules directly. We introduce *TabProbe*, a probing-based algorithm that instantiates this framework, and demonstrate its generality across multiple model classes, including gradient-boosted trees (XGBoost) (Chen & Guestrin, 2016) and large pretrained foundation models (Grinsztajn et al., 2026;

Table 1: **Sample clinical dataset and association rules.** Sample patient records from the Hepatitis dataset (Gong & Turney, 1983) with association rules extracted by our TabProbe from TabPFN (Hollmann et al., 2025), capturing interpretable patterns relating symptoms and treatments to patient survival.

Age	**Sex**	**Steroid**	**Antiviral**	$\cdots$	**Class**
30	female	no	yes	$\cdots$	LIVE
50	male	no	yes	$\cdots$	LIVE
78	male	yes	yes	$\cdots$	LIVE
39	male	yes	no	$\cdots$	DIE
Rule 1	Antiviral = yes $\wedge$ Malaise = yes $\rightarrow$ Class = LIVE				
Rule 2	Malaise = yes $\wedge$ Anorexia = yes $\rightarrow$ Class = LIVE				

Qu et al., 2025; Ma et al., 2024), demonstrating that the framework is not tied to any specific model architecture. As a particularly powerful instantiation, we leverage *Tabular Foundation Models (TFMs)* (Hollmann et al., 2025), large models pretrained on diverse tabular data with strong in-context generalization (Müller et al., 2022), which can be applied *out-of-the-box* without additional task-specific training, directly addressing both rule explosion and the low-data limitations of existing neural approaches. TFMs have demonstrated strong performance as classifiers for small tabular data by pretraining on large-scale synthetic (Qu et al., 2025) or real (Spinaci et al., 2025) tabular data and transferring learned representations to unseen datasets. Despite their growing success, TFMs have not yet been exploited for ARM. Samples from the clinical Hepatitis (Gong & Turney, 1983) dataset, together with association rules learned with TabPFN (Hollmann et al., 2025) using TabProbe, are shown in Table 1.

Our contributions are the following: (1) **A model-agnostic framework** for association rule learning, grounded in a novel framing of ARM as a conditional probability estimation problem, formally decoupling rule learning from frequent itemset mining and enabling any conditional probabilistic model to learn rules directly (Section 3); (2) **Two instantiations of the framework**: *TabProbe*, a novel probing-based algorithm for any single-target conditional probabilistic model, demonstrated with TFMs and gradient-boosted trees; and a multi-target instantiation for reconstruction-based models such as Autoencoders (Bank et al., 2023), together demonstrating the generality of the framework across model classes (Section 3.1); (3) **A comprehensive evaluation** on small ($\sim$100 rows) and larger ($\geq$1000 rows) tabular datasets in two settings: standard rule quality evaluation and downstream classification with cross-validation as part of interpretable rule-based classifiers (Liu et al., 1998; Angelino et al., 2018). With up to 7 times fewer rules, TFMs achieve effectively full data coverage, up to 15% higher association strength (Yan et al., 2009) than state-of-the-art, better predictive performance on unseen data, and remain robust in low-data settings (Section 4).

2 Preliminaries and Related Work

Association Rule Mining. The ARM problem originally seeks rules of the form $X \rightarrow Y$ over categorical data that satisfy user-given minimum *support* and *confidence* thresholds (Agrawal et al., 1994). Support is the proportion of data instances containing $X \cup Y$. Confidence is the conditional frequency of Y among instances containing X.

Rule explosion. The ARM literature has been dominated by algorithmic methods, most notably the FP-Growth (Han et al., 2000) algorithm that has many variations, including parallel FP-Growth (Li et al., 2008) and FP-Growth on GPU (Jiang & Meng, 2017). Such methods work in two stages: finding frequent itemsets of higher support by co-occurrence counting and then forming rules of higher confidence based on given thresholds. However, the algorithmic methods suffer from rule explosion, that is, an excessive number of rules mined that are hard to interpret and post-process (Moens et al., 2013). This is due to the combinatorial growth in both the number of items, i.e., unique values, and their potential combinations into frequent itemsets and rules. The common approaches to overcome rule explosion in ARM include applying item constraints (Srikant et al., 1997; Baralis et al., 2012; Yin et al., 2022), mining only top-k rules (Fournier-Viger et al., 2012; Nguyen et al., 2018), closed itemset mining (Zaki & Hsiao, 2002), which refers to mining

only frequent itemsets without frequent supersets of equal support, constraint programming (Belaid et al., 2019) and SAT-based enumeration (Izza et al., 2021). Beyond such search space reduction methods, recent neurosymbolic alternatives to addressing rule explosion leverage compact learned data representations.

Neurosymbolic methods in ARM. Despite the wide-scale success of deep learning, neural networks have only recently been applied to ARM. Aerial+ (Karabulut et al., 2025b) uses an undercomplete denoising autoencoder to learn lower-dimensional data representations and reconstruct the original inputs. Association rules are then derived based on reconstruction success, with the undercomplete structure favoring statistically significant rules over exhaustive enumeration. There have been other neural network-based approaches to ARM (Berteloot et al., 2024; Patel & Yadav, 2022). To the best of our knowledge, Aerial+ is the *only method* that explicitly addresses rule explosion. However, earlier work showed that in the small-data regime, Aerial+ suffers from insufficient training data, leading to rules that do not accurately capture data patterns (Karabulut et al., 2025a).

Tabular foundation models. Prior-data fitted networks (PFNs) (Müller et al., 2022) pioneered training neural models on large collections of synthetic tabular tasks to enable in-context generalization and serve as a precursor to tabular foundation models. Tabular foundation models (TFMs) extend this paradigm by pretraining large-scale architectures on synthetic or real tabular data to learn transferable representations (Hollmann et al., 2025). Representative TFMs include TabPFN (Hollmann et al., 2025), trained on vast synthetic priors, TabICL (Qu et al., 2025), which broadens in-context learning across larger tables, and TabDPT (Ma et al., 2024), which leverages large-scale real tabular data. Despite the spreading success of TFMs across tabular tasks, use of TFMs to ARM has been limited to fine-tuning Aerial+ with embeddings from TabPFN (Karabulut et al., 2025a).

Our approach. While recent neurosymbolic methods address rule explosion using neural networks, they suffer from poor performance on small datasets, leading to rules that do not capture true patterns in the data. We reframe ARM as a conditional probability estimation problem and introduce a model-agnostic framework that formally decouples discovering association rules from frequent itemset mining. We demonstrate its generality across model classes from gradient-boosted trees to large pretrained foundation models. Using this framing, we show that TFMs can be used for categorical ARM *out-of-the-box* and that they address both rule explosion and low-data scenarios.

3 Model-Agnostic Rule Learning Framework

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a finite set of m items, and $D = \{t_1, t_2, \ldots, t_n\}$ a dataset of transactions, where each transaction $t_i \subseteq I$. For tabular data with k categorical features $F = \{f_1, \ldots, f_k\}$, each feature f_j admits a finite set of c_j categories $\{f_j^1, \ldots, f_j^{c_j}\}$. The item universe is then

$$I = \bigcup_{j=1}^{k} \{f_j^1, \ldots, f_j^{c_j}\}, \tag{1}$$

and the number of items is $m = \sum_{j=1}^{k} c_j$. Each transaction can be represented as a one-hot encoded vector $\mathbf{x} \in \{0,1\}^m$ satisfying $\sum_{i=1}^{c_j} x[f_j^i] = 1$ for all $f_j \in F$.

Example. Table 1 shows a fragment of a clinical dataset where each patient record corresponds to a transaction containing items such as `Sex=male`, `Antiviral=yes`, or `Class=LIVE`. Rules such as

$$\texttt{Antiviral=yes} \wedge \texttt{Malaise=yes} \rightarrow \texttt{Class=LIVE}$$

capture interpretable associations between symptoms, treatments, and outcomes.

Definition 1 (Association Rule)**.** *An association rule is an implication $X \rightarrow Y$, where $X, Y \subseteq I$, is a first-order Horn clause with $|Y| = 1$, $|X| \geq 1$, and $X \cap Y = \varnothing$. The set X is referred as the antecedent and Y is the consequent.*

Classical rule *mining* methods enumerate frequent itemsets and apply thresholding on support and confidence, often leading to a combinatorial explosion of rules. In contrast, we consider association rule *learning* from trained probabilistic models that capture statistical regularities in the data.

Definition 2 (Conditional Probabilistic Model)**.** *Let $\mathcal{M}_\theta$ be a trained model that allows computing conditional queries of the form*

$$P_\theta(i \mid X), \quad i \in I,\ X \subseteq I, \tag{2}$$

interpreted as the model's estimated probability that item i holds given all items in X are observed to hold.

Example. Given the antecedent $X = \{\texttt{Antiviral=yes}, \texttt{Malaise=yes}\}$ from Table 1, the model $\mathcal{M}_\theta$ can be queried for $P_\theta(\texttt{Class=LIVE} \mid X)$.

We *formalize* association rule learning from such models $\mathcal{M}_\theta$ via two requirements.

Requirement 1 (Antecedent Validation)**.** *A model $\mathcal{M}_\theta$ provides a scoring function $s_\theta : 2^I \to [0,1]$, where $s_\theta(X)$ quantifies how plausible the partial configuration X is under the learned data distribution. An antecedent X is considered valid if $s_\theta(X) \geq \tau_a$, for a user-specified threshold $\tau_a \in [0,1]$.*

Intuitively, this prevents the extraction of rules based on unlikely or spurious feature-value combinations.

Requirement 2 (Consequent Extraction)**.** *Given a valid antecedent X, the model allows computing conditional probabilities $P_\theta(i \mid X)$ for all $i \in I \setminus X$. An item i is accepted as a consequent of X if $P_\theta(i \mid X) \geq \tau_c$, for a user-specified threshold $\tau_c \in [0,1]$.*

Example. Let $X = \{\texttt{Antiviral=yes}, \texttt{Malaise=yes}\}$. Suppose the model assigns score $s_\theta(X) = 0.81$, and the antecedent threshold is set to $\tau_a = 0.6$, so that X is accepted as a valid antecedent. Querying the model for conditional probabilities yields:

$$P_\theta(\texttt{Class=LIVE} \mid X) = 0.92,\ P_\theta(\texttt{Class=DIE} \mid X) = 0.08.$$

With a consequent threshold of $\tau_c = 0.8$, the rule

$$\texttt{Antiviral=yes} \wedge \texttt{Malaise=yes} \to \texttt{Class=LIVE}$$

is therefore extracted by the framework.

Definition 3 (Association Validity Function)**.** *Given a model $\mathcal{M}_\theta$, the association validity function is*

$$\mathcal{V}_\theta(X \to Y) = \mathbf{1}[s_\theta(X) \geq \tau_a] \cdot \min_{i \in Y} P_\theta(i \mid X), \tag{3}$$

where $X, Y \subseteq I$ and $\mathbf{1}[\cdot]$ is an indicator function equal to one if its argument is true, and 0 otherwise.

Definition 4 (Association Rule Learning)**.** *Given a trained model $\mathcal{M}_\theta$ and thresholds $\tau_a, \tau_c \in [0,1]$, the association rule learning problem is to identify the set of rules*

$$\mathcal{R} = \{X \to Y \mid \mathcal{V}_\theta(X \to Y) \geq \tau_c\}. \tag{4}$$

3.1 Framework Instantiations

Requirements 1 and 2 can be satisfied through different model architectures. We demonstrate two paradigms, instantiated with multiple models, to show the generality of the framework across model classes.

3.1.1 Multi-Target Rule Learning Paradigm

We now show that multi-target predictive models, such as categorical autoencoders, naturally instantiate the proposed framework. In this setting, a model is trained to reconstruct, predict, or decode a data instance from a possibly corrupted, masked, or noisy input. A *multi-target instantiation* refers to any such model that simultaneously outputs predictions for all features and satisfies the framework requirements. Although not explicitly trained for rule learning, such models can be queried post hoc to extract rules by interpreting their outputs as conditional predictions under partial evidence.

Definition 5 (Multi-Target Instantiation)**.** *Let $\mathcal{M}_\theta : \mathbb{R}^m \to \mathbb{R}^m$ be a trained multi-target model that maps an input vector $\mathbf{x}$ to a reconstructed output $\hat{\mathbf{x}} = \mathcal{M}_\theta(\mathbf{x})$. We assume that the output is normalized per feature, such that for each feature $f_j \in F$, $\sum_{i \in f_j} \hat{\mathbf{x}}[i] = 1$, so that $\hat{\mathbf{x}}[i]$ is the predicted probability of item i.*

Algorithm 1 Multi-Target Rule Learning

Require: Dataset D, model $\mathcal{M}_\theta$, thresholds τ_a, τ_c

1: $\mathcal{R} \leftarrow \emptyset$
2: **for all** antecedents of interest $X \subseteq I$ **do**
3: $\quad \mathbf{x} \leftarrow \text{ConstructProbe}(X)$
4: $\quad \hat{\mathbf{x}} \leftarrow \mathcal{M}_\theta(\mathbf{x})$
5: $\quad$ **if** $\hat{\mathbf{x}}[i'] \geq \tau_a, \forall i' \in X$ **then**
6: $\quad\quad Y \leftarrow \{i \in I \setminus X \mid \hat{\mathbf{x}}[i] \geq \tau_c\}$
7: $\quad\quad \mathcal{R} \leftarrow \mathcal{R} \cup \{X \rightarrow y \mid y \in Y\}$
8: **return** $\mathcal{R}$

Given an antecedent set $X \subseteq I$, we construct an input probe vector $\mathbf{x}(X) \in \mathbb{R}^m$ where $\mathbf{x}(X)[i'] = 1$ for all $i' \in X$, $\mathbf{x}(X)[j] = 0$ for all other items j in features containing items from X, and $\mathbf{x}(X)[k] = 1/|f|$ for items k in remaining features f. We define the induced conditional probability as

$$P_\theta(i \mid X) := \mathcal{M}_\theta(\mathbf{x}(X))[i], \forall i \in I. \tag{5}$$

Example. Let $X = \{\texttt{Antiviral=yes}, \texttt{Malaise=yes}\}$. A sample input probe for $P_\theta(\texttt{Class=LIVE} \mid X)$ would be $p = x(X) = [1, 0, 1, 0, 0.5, 0.5]$, where $p[1], p[3] = 1$ *marks* items in X, $p[2], p[4] = 0$ marks other items in features containing X, and $p[5], p[6] = 0.5$ refers to remaining items. **Note** that other probing strategies are possible depending on model architecture, e.g., leaving $p[5], p[6]$ missing to indicate no prior knowledge.

This instantiation satisfies the two requirements as follows.

- **Antecedent validation (Req. 1).** We define the antecedent score as $s_\theta(X) := \min_{i \in X} P_\theta(i|X)$, which quantifies how plausible the partial configuration X is under the learned data distribution.
- **Consequent extraction (Req. 2).** For a valid antecedent X, item $i \in I \setminus X$ is extracted as a consequent if $P_\theta(i \mid X) \geq \tau_c$.

Aerial+ (Karabulut et al., 2025b) instantiates this paradigm, allowing *simultaneous* antecedent validation and consequent extraction in a single forward pass, as the entire item space is reconstructed at once.

Algorithm 1 illustrates how multi-target association rule learning works in practice. Lines 2-7 iterate over antecedents of interest, e.g., all 3 combinations of $i \in I$. Line 3 constructs an input probe of size m as described earlier. Line 4 performs a forward pass to obtain reconstruction probabilities per feature. Lines 5-6 validate antecedents and extract consequents using thresholds τ_a and τ_c, and line 7 stores the rules.

3.1.2 Single-Target Rule Learning Paradigm

We now consider predictive models that are trained under a feature-wise single prediction objective, *single-target instantiation*, where each feature is predicted from the remaining ones. These models provide access to conditional predictions of the form $P_\theta(i \mid I \setminus f_j)$ for items i belonging to feature f_j. By aggregating such predictions across all features, we obtain a vector $\hat{\mathbf{x}}$ that serves as an approximation of the full conditional prediction vector required by the framework.

Definition 6 (Single-Target Instantiation)**.** *Let $\mathcal{M}_\theta$ be a predictive model that, given an input vector $\mathbf{x}$ and a target feature index j, outputs a probability distribution over the categories of feature f_j, $\mathcal{M}_\theta(\mathbf{x}, j) \in [0,1]^{|f_j|}$. To obtain a prediction vector over all items, we construct*

$$\hat{\mathbf{x}}[i] := \mathcal{M}_\theta(\mathbf{x}_{\setminus f_j}, j)[i], \quad \text{for all } i \in f_j,\ j = 1, \ldots, k, \tag{6}$$

where $\mathbf{x}_{\setminus f_j}$ denotes the input with feature f_j masked or removed to prevent information leakage. Aggregating these predictions over all features yields a vector $\hat{\mathbf{x}} \in [0,1]^m$ that serves as an approximation of the full joint conditional prediction vector, consistent with the feature-wise prediction objective of single-target models.

Algorithm 2 TabProbe: Rule extraction from single-target models

Require: Dataset D, single-target model $\mathcal{M}_\theta$, max antecedents a, thresholds τ_a, τ_c

1: $\mathcal{R} \leftarrow \emptyset$, $F \leftarrow \text{GetFeatures}(D)$
2: **for** $i \leftarrow 1$ to a **do**
3: **for all** feature combination $S \in \binom{F}{i}$ **do**
4: $\mathbf{Q} \leftarrow \text{GenerateProbeVectors}(S, F)$, $\hat{\mathbf{P}} \leftarrow \mathbf{0}^{|\mathbf{Q}| \times m}$
5: **for all** feature $f_j \in F$ **do**
6: $\mathbf{X}^{-j} \leftarrow \text{RemoveFeature}(D, f_j)$
7: $\mathbf{y}^j \leftarrow \text{GetLabels}(D, f_j)$
8: $\mathcal{M}_\theta.\text{fit}(\mathbf{X}^{-j}, \mathbf{y}^j)$
9: $\mathbf{Q}^{-j} \leftarrow \text{RemoveFeature}(\mathbf{Q}, f_j)$
10: $\hat{\mathbf{P}}[\cdot, f_j] \leftarrow \mathcal{M}_\theta.\text{predict_proba}(\mathbf{Q}^{-j})$
11: **for all** probe $\mathbf{q}_r \in \mathbf{Q}$ **do**
12: **if** $\min_{i \in S} \hat{\mathbf{P}}[r, i] \geq \tau_a$ **then**
13: **for all** $i \in I \setminus S$ where $\hat{\mathbf{P}}[r, i] \geq \tau_c$ **do**
14: $\mathcal{R} \leftarrow \mathcal{R} \cup \{S \rightarrow i\}$
15: **return** $\mathcal{R}$

Satisfaction of framework requirements. This instantiation satisfies the requirements as follows.

- **Antecedent validation (Req. 1).** As in the multi-target case, we define $s_\theta(X) := \min_{i \in X} \hat{x}[i]$, which quantifies how plausible the partial configuration X is under the learned data distribution.
- **Consequent extraction (Req. 2).** For a valid antecedent X, $i \in I \setminus X$ is a consequent if $\hat{\mathbf{x}}[i] \geq \tau_c$.

Example. Let $X = \{\texttt{Antiviral=yes}, \texttt{Malaise=yes}\}$ and target feature $f_j = \texttt{Class}$. Following Definition 6, we remove `Class` from D to prepare input matrix $\mathbf{X}^{-j}$, extract f_j labels as $\mathbf{y}^j$, and *fit* $\mathcal{M}_\theta$. The probe vector Q marks `Antiviral=yes` and `Malaise=yes` with probability 1 and assigns uniform priors to remaining features, similar to multi-target instantiation. Querying $\mathcal{M}_\theta$ with X yields $\hat{x}[\texttt{Class=LIVE}] = 0.92$. With $\tau_c = 0.8$, the rule $\texttt{Antiviral=yes} \wedge \texttt{Malaise=yes} \rightarrow \texttt{Class=LIVE}$ is extracted.

TFMs are a natural candidate for this paradigm as they excel at in-context prediction, accurately predicting features conditioned on others *without re-training*, and remain robust in low-data regimes. We instantiate the single-target formulation with three TFMs, TabPFNv3 (Grinsztajn et al., 2026), TabICLv2 (Qu et al., 2026), and TabDPT (Ma et al., 2024), as well as XGBoost (Chen & Guestrin, 2016), a canonical baseline in tabular data tasks, to demonstrate the generality of the framework across model classes.

Unlike multi-target instantiation, single-target predictive models do not allow explicit *marking* of antecedent features when predicting those same features, as this would introduce information leakage. This is an intentional design choice consistent with the feature-wise prediction objective of single-target models, such as TFMs or XGBoost, and its validity is supported by the empirical results presented in Section 4.

Algorithm 2 illustrates how the single-target formulation can be realized with any single-target model $\mathcal{M}_\theta$. For a given max antecedent length a, lines 2-3 enumerate feature combinations for antecedents S. Line 4, using *GenerateProbeVectors()*, generates a probing matrix Q for the marked features in S. Lines 5-10 iterate over each feature $f_j \in F$: *RemoveFeature()* removes f_j from D to prepare the input matrix $\mathbf{X}^{-j}$ (line 6); *GetLabels()* extracts f_j as target labels $\mathbf{y}^j$ (line 7); line 8 fits $\mathcal{M}_\theta$ on $(\mathbf{X}^{-j}, \mathbf{y}^j)$; *RemoveFeature()* removes f_j from Q to obtain $\mathbf{Q}^{-j}$ (line 9); and line 10 calls *predict_proba()* to obtain a probability distribution over values of f_j, stored in $\hat{P}$. Lines 11-14 iterate over $\hat{P}$, performing antecedent validity (line 12) and consequent extraction (line 13) checks. Antecedent combinations and **probe vectors** are generated similarly to Algorithm 1. Figure 1 illustrates rule learning using Algorithm 2 from TFMs.

Time complexity. Algorithm 2 has time complexity $O(k^{a+1} \cdot (T_{\mathcal{M}_\theta}(n) + c^a_{\max}))$, where k is the number of features, a is the maximum antecedent size (typically 2–4), $T_{\mathcal{M}_\theta}(n)$ is the complexity of fitting and predicting

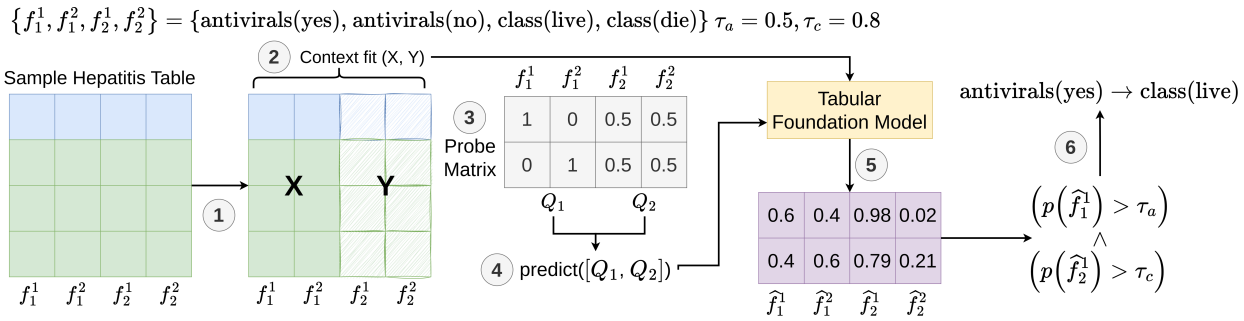

Figure 1: **Example** rule learning from TFMs with Algorithm 2: (1) split table into features (X) and labels (Y), (2) fit TFM on X and Y, (3) construct probing matrix with marked items (probability 1) and uniform priors elsewhere, (4) predict per feature, (5) aggregate predictions into reconstruction matrix, (6) apply thresholding to extract association rules.

with $\mathcal{M}_\theta$ on n rows, and $c_{\max}$ is the maximum categories per feature. $O(k^{a+1})$ is the best achievable complexity for exhaustive antecedent enumeration of length a, as ARM is inherently combinatorial. In practice, a is kept small, making the method polynomial. Note that while Algorithm 2 shows one forward pass per feature for clarity, the implementation performs **a single batched forward pass** over all probing vectors per feature, reducing execution time significantly. See **Appendix A** for a full complexity analysis.

4 Evaluation

This section evaluates our model-agnostic framework instantiations across 2 experimental settings; **i)** the **standard rule quality evaluation** procedure from the ARM literature, **ii) downstream classification evaluation** as part of interpretable machine learning models. Experiments are repeated for 10 random seeds for robustness, and average results are presented. In Setting 2, 5-fold cross-validation across 10 seeds provides a robust estimate of generalization performance, in line with standard evaluation practice in interpretable machine learning (Angelino et al., 2018; Liu et al., 1998). We further provide a scalability analysis in Setting 3 and a hyperparameter analysis in Appendix D.

Baselines. We initialize our model-agnostic framework with 5 different methods: TabPFNv3 (Grinsztajn et al., 2026), XGBoost (Chen & Guestrin, 2016), TabICLv2 (Qu et al., 2026), and TabDPT (Ma et al., 2024) as single-target instantiations and Aerial+ (Karabulut et al., 2025b) as state-of-the-art neurosymbolic baseline, representative of the multi-target instantiation. FP-Growth (Han et al., 2000) is taken as one of the most popular algorithmic ARM baselines, implemented with MLxtend (Raschka, 2018). **Note** that the output of any other algorithmic baseline is identical to FP-Growth (i.e., variants of Apriori (Agrawal et al., 1994), ECLAT (Zaki, 2002) or HMine (Pei et al., 2001)), as they are all deterministic exhaustive enumerators. Furthermore, any search space reduction method from the ARM literature is orthogonal to our framework and directly applicable, as our approach replaces co-occurrence counting rather than the search strategy.

Datasets. 5 small medical tabular datasets (∼100 rows) and 5 larger datasets (300 to 8000 rows) are used, summarized in Table 4 (Appendix B). All datasets have a class label to evaluate downstream classification as part of interpretable ML models (CBA (Liu et al., 1998) and CORELS (Angelino et al., 2018)). Numerical columns are discretized using equal-frequency discretization with 10 bins. For scalability analysis in Section 4.3, the large-scale canonical ARM benchmark T10I4D100K (Agrawal et al., 1994) is used.

Hyperparameters and fair comparison. To ensure a fair comparison, each method is run with a maximum antecedent length of 2. Framework instantiations are run with $\tau_c = 0.8$, and FP-Growth with a minimum confidence threshold of 0.8. In Setting 1, $\tau_a = 0.5$ for all instantiations, while FP-Growth is run with minimum support thresholds $\{0.5, 0.3, 0.2, 0.1\}$, and all results are reported. In Setting 2, $\tau_a = 0.3$ to obtain sufficient classification rules, which tend to have lower support; for the same reason, FP-Growth is run with minimum support thresholds $\{0.5, 0.3, 0.2, 0.1, 0.05, 0.01\}$. All TFMs are run with entire tables as

context and an ensemble size of 8, without any tuning. XGBoost and Aerial+ are tuned per dataset using Optuna (Akiba et al., 2019) and grid search, respectively. FP-Growth has no learnable parameters and is run across all support thresholds. Full parameter settings are listed in Table 7 (Appendix B).

Hyperparameter sensitivity. Appendix D provides an analysis of τ_a and τ_c. Increasing τ_a reduces rule count while increasing support and interestingness, which is analogous to the minimum support threshold. Increasing τ_c raises support, confidence, association strength, and interestingness while reducing rule count. The consistent behavior guides practitioners in threshold selection depending on the rule learning task.

4.1 Setting 1: Rule Quality Evaluation

Setup. In this experimental setting, we follow the standard ARM evaluation practice by computing 5 rule quality metrics on the full dataset, summarized in Table 5 (Appendix B). Zhang's metric measures association strength beyond statistical independence (i.e., non-spuriousness), and interestingness measures how unexpected a rule is relative to its antecedent and consequent supports.

Rule quality evaluation results. Table 2 shows that TFMs achieve full data coverage with up to 7 times fewer rules than FP-Growth, demonstrating that the framework directly addresses rule explosion. TFMs and XGBoost achieve the highest association strength, while Aerial+ and FP-Growth variants consistently show lower Zhang scores despite higher confidence. On small datasets, Aerial+ shows lower support, association strength, and interestingness than TFMs, consistent with its known degradation in low-data regimes. On larger datasets, Aerial+ recovers in coverage and interestingness, but still produces rules of $\sim$15% lower association strength than TFMs and XGBoost. FP-Growth requires lower support thresholds to achieve full data coverage, at the cost of rule explosion; FP-G (0.5) achieves the highest interestingness and support on larger datasets but covers only 92.8% of the data with over 21,000 rules. The only metric where TFMs and XGBoost are consistently lower is confidence; however, as will be discussed below, TFM-learned rules of lower confidence outperform higher-confidence baseline rules on unseen data under 5-fold cross-validation in Setting 2, suggesting that classical methods overfit to dataset-specific co-occurrence frequencies. Figure 2 provides a visual summary via a normalized composite score Q (min-max normalized average of Confidence, Zhang's metric, Interestingness, and Coverage) alongside rule counts; TFMs achieve higher Q with fewer rules than all baselines on both dataset sizes. See **Appendix C.1** for rule quality results per dataset.

Interpretation. Rule quality evaluation leads to 3 core conclusions: i) TFMs can have full data coverage, even in low data settings, with an order of magnitude smaller number of rules than the standard algorithmic rule miners (FP-Growth), addressing rule explosion in ARM, ii) TFM-learned rules show stronger correlations (association strength, Zhang) in both smaller and larger datasets. In larger datasets, TFMs achieve the highest association strength on all datasets, demonstrating practical impact beyond small tabular data. This implies strong non-redundancy in the final rule set, while the majority of the rules learned by algorithmic rule miners will be pruned out due to redundancy. iii) TFMs can address degraded performances in low-data scenarios, which affects the neurosymbolic baseline method Aerial+, as TFMs have up to 15% higher association strength than Aerial+. **Why do TFMs show lower confidence but higher association strength?** TFMs achieve 3–5% lower confidence than FP-Growth and Aerial+, while having up to 15% higher association strength (Zhang). High confidence in classical ARM methods reflects dataset-specific co-occurrence frequencies that do not necessarily generalize to unseen data, whereas association strength measures genuine feature associations beyond statistical independence. A rule capturing a spurious correlation cannot systematically produce higher Zhang scores across datasets, as the metric controls for marginal statistics by construction. This is empirically confirmed in Setting 2, where TFM-learned rules of lower confidence consistently outperform higher-confidence baseline rules on unseen data under 5-fold cross-validation.

4.2 Setting 2: Downstream Task Evaluation

Setup. This experimental setting evaluates all methods as part of two well-known rule-based interpretable ML models: CBA (Liu et al., 1998; Filip & Kliegr, 2018) and CORELS (Angelino et al., 2018). The experimental pipeline is illustrated in Figure 4 (Appendix C.2). CBA and CORELS receive a large set of rules (CBA) or frequent itemsets (CORELS), and select a subset to build a rule-based classifier for high-stakes decision-making. We first learn rules or itemsets with TFMs and the baselines, pass those to CBA and

Table 2: Average rule quality metrics in percentage (%). TFMs and XGBoost achieve the highest association strength (Zhang). TFMs have practically full data coverage with up to 7 times fewer rules than FP-G, which require lower support thresholds for full coverage, leading to rule explosion. Aerial+'s performance degrades (lowest Zhang) in low-data scenarios. (FP-G (x): FP-Growth with min support threshold x).

Algorithm	**#Rules**	**Support**	**Confidence**	**Zhang**	**Interestingness**	**Coverage**
			Small Tabular Datasets			
TabICLv2	331	36.4	82.9	27.2	40.8	97.5
TabPFNv3	274	40.6	86.5	30.4	44.8	97.8
TabDPT	295	36.6	82.9	26.2	40.9	99.7
Aerial+	315	31.4	87.5	22.7	37.1	98.6
XGBoost	292	35.3	83.3	32.7	38.6	99.7
FP-G (0.5)	152	57.8	92.4	23.8	63.2	86.3
FP-G (0.3)	431	42.4	89.8	23.7	48.7	99.7
FP-G (0.2)	680	34.5	90.9	24.4	41.1	100.0
FP-G (0.1)	1463	22.9	91.7	26.0	29.2	100.0
			Larger Tabular Datasets			
TabICLv2	12481	48.2	86.9	34.5	55.1	99.2
TabPFNv3	11885	49.6	88.3	37.4	57.2	98.4
TabDPT	5177	45.8	85.0	36.4	52.9	98.3
Aerial+	7215	50.9	93.6	22.0	57.5	97.6
XGBoost	8711	43.4	83.6	34.1	53.3	89.2
FP-G (0.5)	21825	64.9	91.4	29.4	71.7	92.8
FP-G (0.3)	26538	51.0	91.2	25.9	58.1	98.9
FP-G (0.2)	29211	43.5	91.4	25.2	50.4	99.6
FP-G (0.1)	35376	35.6	91.6	24.4	41.6	100.0

CORELS to build a classifier, and measure accuracy, F1 score, precision, and recall. XGBoost is excluded from this setting as it was included in Setting 1 solely to demonstrate the generality of the framework across model classes, and TFMs outperform XGBoost in rule quality. We use the same parameters as in Setting 1 except $\tau_a = 0.3$ for Aerial+ and TFMs to ensure sufficient classification rules, with dataset-specific reductions detailed in Appendix B.3. For a fair comparison, FP-Growth is evaluated across minimum support thresholds $\{0.3, 0.2, 0.1, 0.05, 0.01\}$, and all results are reported. The experiments are carried out for **10 random seeds** and with **5-fold cross-validation**, and the average results are presented.

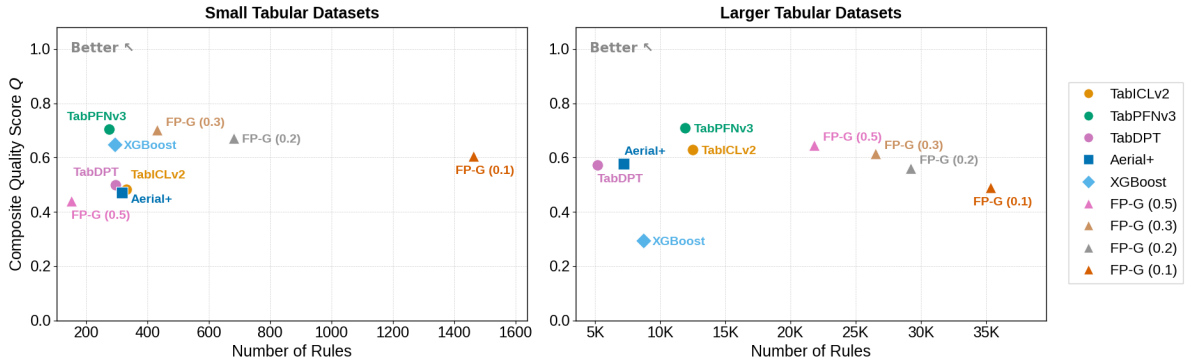

Figure 2: Composite min-max normalized rule quality score $Q = \frac{1}{4}\sum_{m\in\{\text{Conf,Zhang,Inter,Cov}\}} \frac{m-m_{\min}}{m_{\max}-m_{\min}}$, based on Table 2. Support is excluded as its optimal value is application-dependent. TFMs achieve higher Q with fewer rules, confirming their overall rule quality advantage. As individual metrics capture different aspects of rule utility, Q should be interpreted as an exploratory summary rather than a definitive ranking.

Table 3: Predictive performance (%) and average classification rule (CBA) or itemset (CORELS) count averaged across classifiers, with ± denoting standard deviation across random seeds. TFMs outperform all baselines on both small and larger tabular datasets using far fewer rules. FP-Growth is deterministic, hence no variance (FP-G (x): minimum support x). Per-classifier and per-dataset results are in Appendix C.2.

Method	#Rules	Accuracy	F1	Precision	Recall
		Small Tabular Datasets			
Aerial+	267	80.62 ± 1.87	77.33 ± 1.87	76.51 ± 2.04	80.62 ± 1.87
TabPFNv3	145	82.66 ± 3.28	81.04 ± 3.25	81.21 ± 3.67	82.66 ± 3.28
TabICLv2	165	83.32 ± 3.45	81.79 ± 3.53	82.09 ± 3.73	83.32 ± 3.45
TabDPT	221	$\mathbf{83.90 \pm 4.82}$	$\mathbf{82.36 \pm 4.94}$	$\mathbf{82.56 \pm 5.30}$	$\mathbf{83.90 \pm 4.82}$
FP-G (0.3)	57	79.72	76.45	75.84	79.72
FP-G (0.2)	93	79.95	77.08	76.86	79.95
FP-G (0.1)	221	83.54	81.37	82.49	83.54
FP-G (0.05)	506	83.08	80.63	81.42	83.08
FP-G (0.01)	1916	83.28	80.58	81.16	83.28
		Larger Tabular Datasets			
Aerial+	754	87.56 ± 0.78	86.13 ± 0.87	86.34 ± 1.01	87.56 ± 0.78
TabPFNv3	432	87.86 ± 1.48	87.16 ± 1.41	87.46 ± 1.53	87.86 ± 1.48
TabICLv2	420	$\mathbf{88.31 \pm 0.88}$	$\mathbf{87.64 \pm 0.91}$	$\mathbf{88.02 \pm 1.00}$	$\mathbf{88.31 \pm 0.88}$
TabDPT	410	84.94 ± 2.73	84.32 ± 2.75	84.82 ± 2.79	84.94 ± 2.73
FP-G (0.3)	292	84.90	84.36	84.53	84.90
FP-G (0.2)	385	88.10	87.30	87.30	88.10
FP-G (0.1)	617	87.87	86.50	86.33	87.87
FP-G (0.05)	1373	88.13	86.61	86.34	88.13
FP-G (0.01)	2182	88.04	86.60	86.41	88.04

Algorithm 2 is adapted to mine frequent itemsets (Algorithm 3 of **Appendix E**) solely to accommodate CORELS, a canonical rule-based model, as frequent itemset mining is not a requirement of our method.

Results. Table 3 shows that on small datasets, TabDPT and TabICLv2 outperform Aerial+ and FP-Growth using far fewer rules. TabPFNv3 is only 0.6% behind FP-Growth (0.01) in accuracy and recall, while being better in F1 and precision and outperforming Aerial+ with fewer rules. On larger datasets, TabICLv2 outperforms all baselines across all metrics, with TabPFNv3 being 0.5% behind TabICLv2. On both small and larger datasets, TFMs lead to the smallest number of rules with the highest overall predictive performance. Aerial+'s predictive performance recovers on larger datasets but remains slightly behind TabPFNv3 and TabICLv2. FP-Growth leads to higher predictive performance as the min support threshold decreases, at the expense of rule explosion. See **Appendix C.2** for results per dataset and per classifier.

Interpretation. All TFMs address rule explosion in ARM, learning concise rule sets with better predictive performance than baselines on both dataset sizes. In low-data scenarios, Aerial+ degrades significantly while TFMs remain the best-performing option, which confirms the practical advantage of foundation models in low-data settings. On larger datasets, TFMs outperform all FP-Growth configurations despite using far fewer rules, demonstrating their applicability beyond small tabular data. FP-Growth with lower minimum support thresholds produces large numbers of class association rules that do not improve predictive performance under cross-validation, providing further empirical evidence that exhaustive enumeration captures dataset-specific patterns that do not generalize to unseen data. Together with rule quality evaluation, we conclude that TFMs address the degraded performance of neurosymbolic approaches in low-data scenarios, for both knowledge discovery and interpretable high-stakes decision-making. TFM performance can be further improved by reducing τ_a to obtain more rules of lower support (see Appendix D for hyperparameter analysis).

4.3 Setting 3: Scalability Analysis

Setup. TFMs and baselines are run for 5 random seeds with the same setup as Section 4.1, and average execution times are recorded. Experiments are conducted on the first 5,000 rows of the T10I4D100K

dataset (Agrawal et al., 1994), a canonical benchmark for execution time evaluation in association rule mining, with an increasing number of items as the itemset universe is the decisive factor in scalability.

Results. Figure 3 shows mean execution times in seconds as the number of columns increases. TFMs (TabPFNv3, TabICLv2, TabDPT) consistently outperform all FP-Growth variants and scale to 200 columns within the 1,000-second limit, while FP-Growth variants exceed this limit after 120–140 columns. This demonstrates that the scalability concern associated with exhaustive enumeration is more severe in practice for standard rule miners than TFM-based rule learning. Aerial+ remains the fastest method due to its small, specialized architecture. However, TFMs come within a comparable execution time range despite being general-purpose foundation models orders of magnitude larger than Aerial+'s compact autoencoder.

Interpretation. Despite being general-purpose foundation models pretrained on large, diverse tabular data, TFMs achieve execution times close to Aerial+, which has a compact autoencoder architecture specialized for fast tabular reconstruction. This, combined with their superior rule quality and generalization performance demonstrated in Settings 1 and 2, makes TFMs the stronger overall choice for association rule learning. Furthermore, TFMs scale beyond FP-Growth variants, demonstrating that the scalability concern associated with exhaustive enumeration is more severe in practice. Considering the speed of advancement in TFMs, going from tables of up to 10,000 rows (Hollmann et al., 2022; 2025) to 500,000 rows and 2,000 columns (Ma et al., 2024) within 2 years, execution time is expected to improve further, independently of our framework.

5 Discussion

Model-agnostic association rule learning. The framework introduced in Section 3 reframes ARM as a learning problem rather than a search over co-occurrence counts, shifting the question from *how often do these items appear together?* to *what does the learned model believe about their conditional dependency?* This new framing applies to any conditional probabilistic model on tabular data satisfying two requirements: antecedent validation, which checks whether a candidate antecedent configuration is plausible under the learned data distribution, and consequent extraction, which identifies items whose conditional probability exceeds a threshold given that antecedent. Together, these two requirements decouple rule extraction from frequent itemset mining, enabling any model that can estimate conditional probabilities to serve as the backbone of an ARM system. As demonstrated with TFMs, gradient-boosted trees, and reconstruction-based models in Section 3.1, a diverse range of model classes can instantiate the framework, offering practitioners flexibility in choosing a model suited to their data and computational constraints. This framework also opens a new avenue for neural network interpretability, as association rules provide inherently interpretable explanations of learned patterns in an otherwise opaque model.

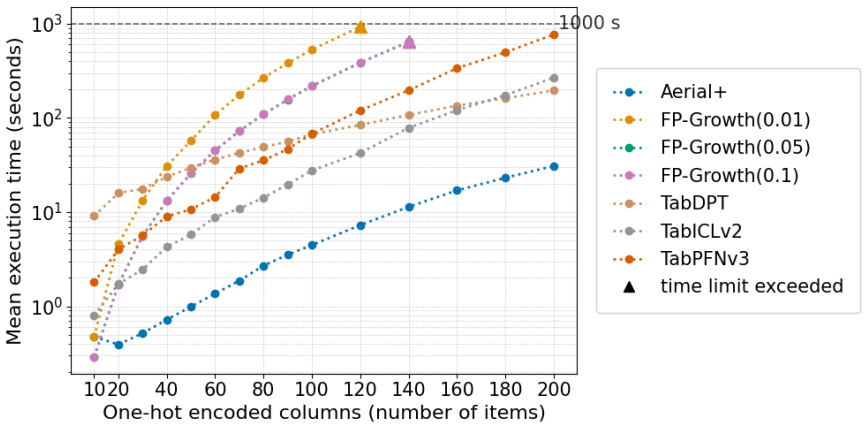

Figure 3: Mean execution time on the canonical T10I4D100K ARM benchmark as the number of columns increases. TFMs scale to 200 columns within the 1,000-second limit while FP-Growth variants exceed the limit after 120–140 columns, and achieve execution times close to Aerial+ despite being general-purpose foundation models of orders of magnitude larger model size (FP-Growth (x): x is min support threshold).

Tabular foundation models can learn association rules. TFMs with a classification objective can be adapted to learn association rules using the single-target instantiation of our framework out-of-the-box without re-training or architecture optimization, in contrast to neurosymbolic methods such as Aerial+, which require per-dataset training and architecture fine-tuning. As shown in Setting 1 (Section 4.1), TFMs produce concise rule sets with effectively full data coverage, addressing rule explosion in ARM on tabular data. Furthermore, TFM-learned rules show stronger generalizability than state-of-the-art methods, achieving higher association strength and better downstream performance on unseen data under cross-validation, as confirmed in Setting 2 (Section 4.2). Notably, lower-confidence TFM rules consistently outperform higher-confidence baseline rules on held-out data, suggesting that classical methods overfit to dataset-specific co-occurrence frequencies while TFMs capture genuine feature associations that generalize to unseen data.

Scalability. As demonstrated in Setting 3 (Section 4.3), TFMs scale beyond FP-Growth in execution time, reaching 200 items within the 1,000-second limit while FP-Growth variants exceed it beyond 120–140 items. As a widely adopted and representative algorithmic baseline, FP-Growth reflects the scalability of co-occurrence counting methods in general; the superior scaling of TFMs therefore suggests a fundamental advantage of the learning-based framing over counting-based ARM as the itemset universe grows. This is particularly notable given that TFMs are general-purpose foundation models orders of magnitude larger than specialized methods such as Aerial+. Feature-wise predictions in TFMs are parallelizable across features, and some TFMs (e.g., TabPFNv3) support context caching to further speed up inference. The remaining scalability constraint is the context size limit of current TFM architectures, although this has improved dramatically, from tables of up to 10,000 rows (Hollmann et al., 2022) to 500,000 rows (Qu et al., 2025) within two years, and is expected to continue improving independently of our framework.

Practical implications. Many concepts from the ARM literature that reduce search space can be directly applied to our framework and Algorithms 1 and 2. These include ARM with item constraints (Yin et al., 2022), where probing vectors are constructed for features of interest only, and top-k rule mining (Fournier-Viger et al., 2012), where rules of top-k highest probability are learned rather than all rules above a threshold. Probing vectors can also be constructed iteratively, limited to features exceeding τ_a in the previous iteration, reducing the probing matrix size and computational cost. As shown in the hyperparameter analysis in Appendix D, τ_a and τ_c allow practitioners to balance rule set size and rule quality according to their needs, with lower τ_a favoring recall and higher τ_c favoring precision in downstream classification. For practitioners without access to GPU resources or large pretrained models, the framework can be instantiated with any lightweight conditional probabilistic model requiring only standard CPU resources, as demonstrated with gradient-boosted trees in our evaluation (Section 4.1).

6 Conclusions and Future Work

This paper introduced a model-agnostic framework that reframes association rule mining as a conditional probability estimation problem, formally decoupling rule extraction from frequent itemset mining. The framework applies to any conditional probabilistic model over tabular data that satisfies two requirements: a scoring mechanism for antecedent validation and conditional probability estimation for consequent extraction. We instantiate the framework with tabular foundation models, gradient-boosted trees, and reconstruction-based models, demonstrating its generality across model classes.

In two comprehensive experimental settings, we demonstrated that tabular foundation model-learned rules achieve full data coverage with stronger associations and less redundancy than state-of-the-art methods, while gradient-boosted trees as a lightweight instantiation further confirm the generality of the framework. Furthermore, these rules generalize to unseen data, achieving equal or higher classification performance within interpretable machine learning pipelines despite using an order of magnitude fewer rules.

This work opens several directions for future research. Evaluating other conditional probabilistic models within the framework is a natural next step. Extending the approach to numerical data without pre-discretization, adapting it to non-tabular structures such as graphs, and applying it to various association rule mining variants, such as sequential rule mining and high-utility itemset mining, are further promising directions. More broadly, our framework shows how pretrained foundation models can be leveraged for interpretable knowledge discovery without task-specific training.

References

Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, volume 1215, pp. 487–499, 1994.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234):1–78, 2018.

Jineta Banerjee, Jaclyn N Taroni, Robert J Allaway, Deepashree Venkatesh Prasad, Justin Guinney, and Casey Greene. Machine learning in rare disease. *Nature Methods*, 20(6):803–814, 2023.

Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

Elena Baralis, Luca Cagliero, Tania Cerquitelli, and Paolo Garza. Generalized association rule mining with constraints. *Information Sciences*, 194:68–84, 2012.

Mohamed-Bachir Belaid, Christian Bessiere, and Nadjib Lazaar. Constraint programming for association rules. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 127–135. SIAM, 2019.

Théophile Berteloot, Richard Khoury, and Audrey Durand. Association rules mining with auto-encoders. In *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 51–62. Springer, 2024.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Jiří Filip and Tomáš Kliegr. Classification based on associations (cba)-a performance analysis. Technical report, EasyChair, 2018.

Philippe Fournier-Viger, Cheng-Wei Wu, and Vincent S Tseng. Mining top-k association rules. In *Advances in Artificial Intelligence: 25th Canadian Conference on Artificial Intelligence, Canadian AI 2012, Toronto, ON, Canada, May 28-30, 2012. Proceedings 25*, pp. 61–73. Springer, 2012.

Ashish Ghosh and Bhabesh Nath. Multi-objective rule mining using genetic algorithms. *Information Sciences*, 163(1-3):123–133, 2004.

Gail Gong and Peter Turney. Hepatitis dataset, 1983. URL `https://archive.ics.uci.edu/dataset/46/hepatitis`.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

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.

Yacine Izza, Saïd Jabbour, Badran Raddaoui, and Abdelhamid Boudane. On the enumeration of association rules: A decomposition-based approach. In *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pp. 1265–1271, 2021.

Hao Jiang and He Meng. A parallel fp-growth algorithm based on gpu. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pp. 97–102. IEEE, 2017.

Erkan Karabulut, Daniel Daza, Paul Groth, and Victoria Degeler. Discovering association rules in high-dimensional small tabular data. In *Proceedings of the 1st International Workshop on Advanced Neuro-Symbolic Applications (ANSyA) at the European Conference on Artificial Intelligence (ECAI 2025)*, 2025a. Accepted/In press.

Erkan Karabulut, Paul Groth, and Victoria Degeler. Neurosymbolic association rule mining from tabular data. In *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning (NeSy 2025)*, volume 284 of *Proceedings of Machine Learning Research*, pp. 565–588. PMLR, 08–10 Sep 2025b. URL `https://proceedings.mlr.press/v284/karabulut25a.html`.

Minakshi Kaushik, Rahul Sharma, Iztok Fister Jr, and Dirk Draheim. Numerical association rule mining: a systematic literature review. *arXiv preprint arXiv:2307.00662*, 2023.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. 2023. URL `https://archive.ics.uci.edu`.

Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y Chang. Pfp: parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 107–114, 2008.

Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD*, pp. 80–86, 1998.

Bo Liu, Ying Wei, Yu Zhang, and Qiang Yang. Deep neural networks for high dimension, low sample size data. In *IJCAI*, volume 2017, pp. 2287–2293, 2017.

José María Luna, Philippe Fournier-Viger, and Sebastián Ventura. Frequent itemset mining: A 25 years review. *WIREs Data Mining and Knowledge Discovery*, 9(6):e1329, 2019. doi: https://doi.org/10.1002/widm.1329. URL `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1329`.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L Caterini. Tabdpt: Scaling tabular foundation models. *arXiv preprint arXiv:2410.18164*, 2024.

Sandy Moens, Emin Aksehirli, and Bart Goethals. Frequent itemset mining for big data. In *2013 IEEE international conference on big data*, pp. 111–118. IEEE, 2013.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=KSugKcbNf9`.

Linh TT Nguyen, Bay Vo, Loan TT Nguyen, Philippe Fournier-Viger, and Ali Selamat. Etarm: an efficient top-k association rule mining algorithm. *Applied Intelligence*, 48:1148–1160, 2018.

Harvendra Kumar Patel and KP Yadav. An innovative approach for association rule mining in grocery dataset based on non-negative matrix factorization and autoencoder. *Journal of Algebraic Statistics*, 13 (3):2898–2905, 2022.

Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang, and Dongqing Yang. H-mine: Hyper-structure mining of frequent patterns in large databases. In *proceedings 2001 IEEE international conference on data mining*, pp. 441–448. IEEE, 2001.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

Sebastian Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, 3(24), April 2018. doi: 10.21105/joss.00638.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. Contexttab: A semantics-aware tabular in-context learner. *arXiv preprint arXiv:2506.10707*, 2025.

Ramakrishnan Srikant, Quoc Vu, and Rakesh Agrawal. Mining association rules with item constraints. In *Kdd*, volume 97, pp. 67–73, 1997.

Xiaowei Yan, Chengqi Zhang, and Shichao Zhang. Confidence metrics for association rule mining. *Applied Artificial Intelligence*, 23(8):713–737, 2009.

Zhaowen Yin, Wensheng Gan, Gengsen Huang, Yongdong Wu, and Philippe Fournier-Viger. Constraint-based sequential rule mining. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. IEEE, 2022.

Mohammed J Zaki and Ching-Jui Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proceedings of the 2002 SIAM international conference on data mining*, pp. 457–473. SIAM, 2002.

Mohammed Javeed Zaki. Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, 12(3):372–390, 2002.

A Runtime Complexity Analysis

This section provides a runtime complexity analysis of Algorithm 2 in big O notation.

Initials. Building on the ARM problem definition given in Section 2 and Definition 1; let k denote the number of features in a tabular data, $c_{\max} = \max_j c_j$ the maximum categories per feature, $m = \sum_{j=1}^{k} c_j$ the total number of items, n the number of transactions in D, a the maximum antecedent size, and $T_{\mathcal{M}_\theta}(n)$ the time complexity of *fitting* and *predicting* with tabular foundation model $\mathcal{M}_\theta$ on n rows.

Component-wise analysis. We analyze each component of the algorithm separately and then aggregate the results.

Outer Loops. Lines 2-14 iterate over antecedent sizes 1 to a. Lines 3-16 generates $\binom{k}{i}$ feature combinations for each antecedent size i, and generates a total of

$$\sum_{i=1}^{a} \binom{k}{i} = O(k^a) \quad \text{combinations for constant } a \tag{7}$$

Probe Vector Generation. For each feature combination S with $|S| = i$ in line 3, we generate probing vectors for all feature category combinations. The number of probing vectors is $\prod_{f \in S} c_f \leq c_{\max}^i$. Each probe has a dimension m. Therefore, generating each probe vectors Q take $O(c_{\max}^i \cdot m)$.

Feature-wise Prediction. Lines 5-10 iterates over the k features and:

- Prepare X_{ctx} context by removing feature f_j from D in line 6, and storing it as y_{ctx} in line 7: $O(1)$
- Fit the foundation model in line 8: $T_{\mathcal{M}_\theta}^{\text{fit}}(n)$
- Remove f_j from Q and predict on all remaining Q^{-j} probing vectors: $T_{\mathcal{M}_\theta}^{\text{pred}}(c_{\max}^i)$

Total complexity of feature-wise predictions for k features becomes: $O(k \cdot (T_{\mathcal{M}_\theta}(n) + T_{\mathcal{M}_\theta}(c_{\max}^i)))$

Rule Extraction. Lines 11-14 extract rules from the final prediction matrix $\hat{P}$ by checking each k features for c_{max}^i probes in $O(k \cdot c_{max}^i)$.

Total Complexity. Aggregating complexities across all components:

$$\sum_{i=1}^{a} \binom{k}{i} \cdot [k \cdot (T_{\mathcal{M}_\theta}(n) + T_{\mathcal{M}_\theta}(c_{max}^i)) + k \cdot c_{max}^i] \tag{8}$$

Assuming $n \gg c_{max}$ for a typical tabular data and a being constant in practice:

$$T_{\text{total}} = O\left(k^{a+1} \cdot T_{\mathcal{M}_\theta}(n) + k^{a+1} \cdot c_{\max}^a\right) \tag{9}$$

Thus, Algorithm 2 has time complexity $O(k^{a+1} \cdot (T_{\mathcal{M}_\theta}(n) + c_{\max}^a))$ where k is the number of features, a is the maximum antecedent size (typically 2-4), n is the dataset size, and $c_{\max}$ is the maximum classes per feature.

Compared to Aerial+, the single-target rule learning paradigm, instantiated with TFMs, requires k forward passes per probe (one per feature), resulting in a factor of k slowdown during rule extraction. However, when using pretrained foundation models, it eliminates the training phase, which can be advantageous for small datasets or rapid deployment.

B Experimental Details

The experiments are run on a machine with an NVIDIA H100 GPU with 94 GiB memory, and a 4-core AMD EPYC 7F72 CPU per task.

This section includes an extended version of the Evaluation section with finer-grained results.

Open-source repository. Source code of all the experiments, including re-usable functions and the datasets, is openly available at https://anonymous.4open.science/r/tabprobe-D0C0.

B.1 Datasets

10 datasets from the canonical UCI ML repository (Kelly et al., 2023) are used across Settings 1 and 2, summarized in Table 4. 5 small medical tabular datasets (∼100 rows) evaluate TFMs in low-data scenarios, while 5 larger datasets (300 to 8000 rows) emulate scenarios with sufficient data to train ML models. All datasets have a class label to evaluate downstream classification as part of interpretable ML models (CBA (Liu et al., 1998) and CORELS (Angelino et al., 2018)). Numerical columns are discretized using equal-frequency discretization with 10 bins. CSV versions of the datasets are available in the *data* folder of our repository.

Table 4: Datasets from the canonical UCI ML repository (Kelly et al., 2023) used in Settings 1 and 2.

Dataset	**Columns**	**Items**	**Rows**
Small Tabular Datasets			
Acute Inflammations	6	22	120
Hepatitis	19	89	155
Cervical Cancer	19	197	72
ASD Screening (Adolescent)	21	95	104
Fertility	9	36	100
Larger Tabular Datasets			
Breast Cancer	9	43	286
Congressional Voting Records	16	34	435
Mushroom	22	118	8124
Chess (KR vs. KP)	35	73	3196
Spambase	57	157	4601

B.2 Rule Quality Metrics

Table 5 defines the 5 rule quality metrics used in Setting 1. Zhang's metric (Yan et al., 2009) measures genuine association strength beyond statistical independence, controlling for marginal statistics by construction. Interestingness (Ghosh & Nath, 2004) measures how unexpected a rule is relative to its antecedent and consequent supports.

Table 5: Rule quality metrics for a rule $X \to Y$ over dataset D.

Metric	**Formula**	**Range**
Support	$\lvert t \in (X \cup Y)\rvert / \lvert D\rvert$	$[0, 1]$
Confidence	$\sup(X \to Y)/\sup(Y)$	$[0, 1]$
Coverage	$\lvert\{t \mid \exists(X \to Y) \in \mathcal{R} : X \subseteq t\}\rvert / \lvert D\rvert$	$[0, 1]$
Zhang's (Yan et al., 2009)	$\frac{\text{conf}(X\to Y)-\text{conf}(X'\to Y)}{\max(\text{conf}(X\to Y),\text{conf}(X'\to Y))}$	$[-1, 1]$
Interestingness (Ghosh & Nath, 2004)	$\frac{\sup(X\to Y)}{\sup(X)} \cdot \frac{\sup(X\to Y)}{\sup(Y)} \cdot (1 - \frac{\sup(X\to Y)}{\lvert D\rvert})$	$[0, 1]$

B.3 Reproducibility and Parameter Settings

Fixed execution seeds. All experiments are repeated with the following 10 fixed seeds, listed in Table 6.

Table 6: Fixed execution seeds for reproducibility.

Seeds
42, 1608637542, 1273642419, 1935803228, 787846414, 996406378, 1201263687, 423734972, 415968276, 670094950

Parameter settings. Full parameter settings for all framework instantiations and baselines are listed in Table 7.

Table 7: Parameter settings for framework instantiations and baselines across experimental settings. a is the antecedent length. $^{\dagger}\tau_a = 0.5$ for Setting 1, $\tau_a = 0.3$ for Setting 2 to obtain sufficient classification rules. ‡min_supp $\in \{0.5, 0.3, 0.2, 0.1\}$ for Setting 1, $\in \{0.3, 0.2, 0.1, 0.05, 0.01\}$ for Setting 2, as classification rules tend to have lower support.

Method	Parameters
TabPFNv3	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, n_estimator $= 8$
TabICLv2	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, n_estimator $= 8$
TabDPT	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, n_estimator $= 8$
Aerial+	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, grid search-tuned per dataset
XGBoost	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, Optuna-tuned per dataset
FP-Growth	$a = 2$, min. confidence= 0.8, min. support $\in \{0.5, 0.3, 0.2, 0.1\}^{\ddagger}$

Dataset-specific parameter adjustments. In Setting 2, τ_a is further reduced for specific datasets while running CBA to obtain sufficient classification rules of lower support. For *cervical cancer behavior risk* and *breast cancer* datasets, $\tau_a = 0.1$ is used.

Aerial+ architecture parameters. Aerial+ utilizes an undercomplete denoising autoencoder whose architecture is optimized per dataset via grid search. The resulting parameters are listed in Table 8; all other parameters are kept at default values.

Table 8: Optimized Aerial+ architecture and training parameters per dataset.

Dataset	Batch Size	Layers	Epochs
Small Tabular Datasets			
Acute Inflammations	2	4	10
Hepatitis	2	4	10
Cervical Cancer	1	8	20
ASD Screen. (Adolescent)	2	4	10
Fertility	2	4	10
Larger Tabular Datasets			
Breast Cancer	2	4	5
Congressional Vot. Recs.	4	2	2
Mushroom	64	64	2
Chess (KR vs. KP)	8	[16, 4]	5
Spambase	64	64	2

C Extended Experimental Results

This section presents extended experimental results for experimental setting 1 (Section 4.1) and setting 2 (Section 4.2). Please refer to those sections for the definitions and metrics used in each setting.

C.1 Extended Rule Quality Evaluation

Table 9 shows rule quality evaluation results for TFMs and baselines on small tabular datasets. The results show that even small tabular data may lead to rule explosion, as FP-Growth resulted in 4088 rules when run on the Autism Screening dataset with 10% min support threshold. Increasing the min support threshold reduces the number of rules significantly; however, it also reduces the association strength of the rules learned. TFMs and Aerial+, however, can address rule explosion as they lead to full data coverage over 4 out of 5 datasets, and close to 100% in the Cervical Cancer Behavior Risk dataset. Aerial+ produced rules of lower association strength in 4 out of 5 datasets, except Hepatitis, where Aerial+ and TabPFNv3 have the same association strength. In terms of the interestingness of the rules, TFM-learned rules outperformed Aerial+ on all datasets. The confidence score of TFMs is consistently lower than the baselines. We argue, and experimentally showed in setting 2 (see Section 4.2), that this is due to TFM-learned rules capturing more generalizable patterns as opposed to dataset-specific idiosyncrasies. This is partially implied by higher association strength and interestingness scores in TFM rules compared to baselines. The results in setting 2 showed that TFM-learned rules outperform the baselines when applied to unseen data in classification tasks.

Table 10 shows rule quality evaluation results for larger tabular datasets. On these datasets, the impact of rule explosion is more visible, e.g., as FP-Growth produces more than 100.000 rules on the Spambase dataset even with 30% min support threshold. Both TFMs and Aerial+ were again able to learn much more compact rule sets with higher statistical significance (Zhang and interestingness) than FP-Growth. The confidence scores of TFMs were relatively higher in large datasets than in small datasets when compared to baselines. TFM-learned rules have the highest association strength on all datasets and full data coverage in 4 out of the 5 datasets.

C.2 Extended Downstream Task Evaluation

This section presents extended downstream task evaluation results of TFMs and baselines per dataset (see Section 4.2).

Experimental setup for downstream task evaluation is shown in Figure 4. CBA Liu et al. (1998) and CORELS Angelino et al. (2018) are two well-known interpretable ML models used in high-stakes decision-making. CBA receives a set of association rules while CORELS receives a set of frequent itemsets to select and order a subset to build a rule-based classifier. Each tabular dataset is first split into training and test folds for 5-fold cross-validation. TFMs and baselines are run on the training folds to learn association rules for CBA and frequent itemsets for CORELS (see Section E for an adaptation of Algorithm 2 to frequent itemset learning). Rule-based classifiers are then built using CBA and CORELS and evaluated on the test folds based on accuracy, precision, recall, and F1 score.

The results of classification tasks for TFMs and baselines per dataset are given in Figures 5 and 6. With CORELS, TFMs outperform all baselines on both small and larger tabular datasets. On small datasets, TFMs achieve better or comparable ($\pm 2\%$) performance across all datasets, while Aerial+ falls behind FP-Growth on Autism Screening and Cervical Cancer. On larger datasets, TFMs again lead across all datasets, with Aerial+ recovering but remaining $\sim$1% behind TabICLv2 and TabPFNv3. With CBA, TFMs have better or comparable performance in 3 out of 5 small datasets and 4 out of 5 larger datasets.

C.3 Interpretation of the Overall Evaluation Results

Rule quality, downstream tasks, and scalability evaluations together imply the following 3 conclusions:

1. **TFMs can address rule explosion in ARM.** Our approach can utilize TFMs out-of-the-box to learn a concise set of association rules with high data coverage and high statistical rule quality similar to Aerial+.

2. **TFMs remain robust in low-data scenarios for association rule learning.** While the performance of Aerial+ degrades on small tabular data in terms of association strength (Zhang), in-

terestingness, and predictive performance, TFMs remain robust in such scenarios and outperform Aerial+ and FP-Growth in rule quality as well as in predictive performance.

3. **TFM-based rule learning is scalable.** TFMs scale to an itemset universe size of 200 within the 1,000-second limit, outperforming all FP-Growth variants which exceed the limit after 120–140 items, while achieving execution times close to Aerial+ despite having orders of magnitude larger model sizes. This is due to the fact that itemset universe size is the decisive factor in scalability rather than model size. As TFM architectures continue to advance, from supporting up to 10,000 rows (Hollmann et al., 2022; 2025) to 500,000 rows and 2,000 columns (Ma et al., 2024) within two years, execution time is expected to improve further, independently of the rule learning framework itself.

D Hyperparameter Analysis

This section analyzes the hyperparameters of Algorithm 2.

Setup. Each TFM is run on all 10 datasets with 10 random seeds, varying τ_a and τ_c thresholds. First, $\tau_c = 0.8$ is fixed and $\tau_a \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ is varied. Next, $\tau_a = 0.5$ is fixed and $\tau_c \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ is varied. Algorithm 2 is executed per TFM-dataset pair up to line 12 to obtain $\hat{P}$, and the thresholds are applied to the same $\hat{P}$, ensuring comparable results across hyperparameter configurations.

The results of this analysis is presented in Figure 7. This figure shows the change in average rule counts and rule quality scores (support, confidence, Zhang's metric, and interestingness) as τ_a and τ_c change per TFM. The first row shows that as τ_a increases, rule count decreases while support and interestingness scores increase. No significant change is observed in confidence and association strength (Zhang) scores. The second row shows that as τ_c increases, support, confidence, association strength, and interestingness scores increase significantly, while rule count decreases.

The implications of the results are that τ_a impacts rule quality synonymously to the minimum support threshold of algorithmic methods, while τ_c has a more complex impact than the minimum confidence threshold. Hyperparameter settings depend on the rule learning task, as rules of different quality may lead to different results across tasks. As an example, in an anomaly detection task, when considered as a classi-

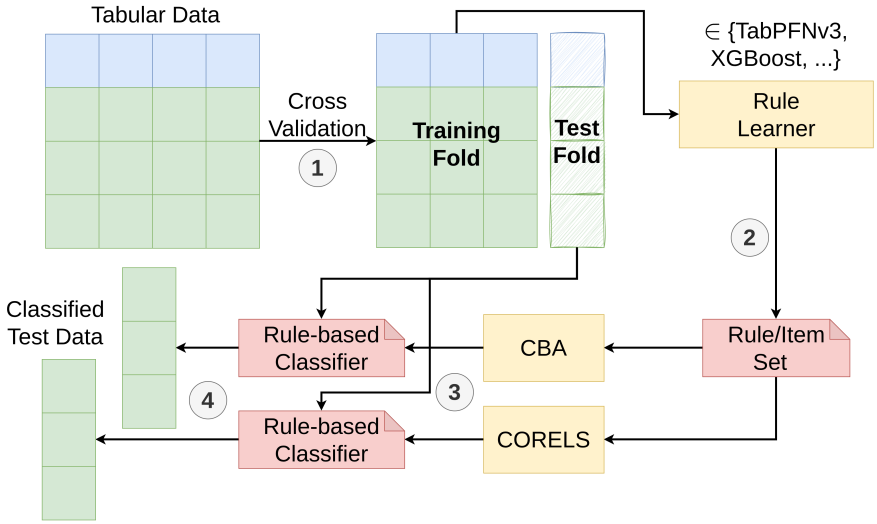

Figure 4: Experimental Setting 2 for downstream task evaluation. (1) Tabular data is split into training and test folds for cross-validation, (2) TFMs and baselines learn rules (CBA) or frequent itemsets (CORELS) on the training fold, (3) learned rules and itemsets are passed to CBA and CORELS to build a rule-based classifier, (4) classifiers are evaluated on the test fold. This process is repeated for 5-fold across 10 seeds.

Algorithm 3 Frequent itemset mining with single-target models

Require: Dataset D, single-target model $\mathcal{M}_\theta$, max antecedents a, similarity threshold τ_s
Ensure: Frequent itemsets $\mathcal{I}$

1: $\mathcal{I} \leftarrow \emptyset$
2: $F \leftarrow$ GetFeatures(D)
3: **for** $i \leftarrow 1$ to a **do**
4: **for all** feature combination $S \in \binom{F}{i}$ **do**
5: $\mathbf{Q} \leftarrow$ GenerateProbeVectors(S, F)
6: $\hat{\mathbf{P}} \leftarrow \mathbf{0}^{|\mathbf{Q}| \times m}$
7: **for all** feature $s_j \in S$ **do**
8: $\mathbf{X}^{-j} \leftarrow$ RemoveFeature(D, f_j)
9: $\mathbf{y}^j \leftarrow$ GetLabels(D, f_j)
10: $\mathcal{M}_\theta$.fit($\mathbf{X}^{-j}, \mathbf{y}^j$)
11: $\mathbf{Q}^{-j} \leftarrow$ RemoveFeature($\mathbf{Q}, f_j$)
12: $\hat{\mathbf{P}}[\cdot, f_j] \leftarrow \mathcal{M}_\theta$.predict_proba($\mathbf{Q}^{-j}$)
13: **for all** probe $\mathbf{q}_r \in \mathbf{Q}$ **do**
14: **if** $\min_{i \in S} \hat{\mathbf{P}}[r, i] \geq \tau_s$ **then**
15: $\mathcal{I} \leftarrow \mathcal{I} \cup \{S\}$
16: **return** $\mathcal{I}$

fication task, it can be carried out with a rule-based classifier that should include rules of lower support. Therefore, lower τ_a values should be preferred. On the contrary, in a knowledge discovery task with the goal of finding interesting patterns in the data, both τ_a and τ_c should be set sufficiently high to optimize rule interestingness and association strength, while avoiding an excessively small number of rules.

E Frequent Itemset Mining with Tabular Foundation Models

As part of experimental setting 2 (Section 4.2), we utilized CORELS Angelino et al. (2018) to build rule-based classifiers for downstream task evaluation besides CBA for robust results. CORELS require frequent itemsets to build classifiers. Therefore, this section provides an adaptation of Algorithm 2 to learn frequent itemsets with TFMs.

The algorithm to learn frequent itemsets with single-target models, i.e., TFMs, out-of-the-box is given in Algorithm 3. Differences to Algorithm 2 are marked with a green background color. Instead of two probabilistic antecedent and consequent thresholds, τ_a and τ_c, we now have a single similarity threshold τ_s. In the loop between lines 7-12, the algorithm queries TFMs per feature in S only and not all the features in F. In lines 14-15, instead of threshold checks on antecedents and consequents separately, the τ_s threshold is checked against the marked features in S. The feature sets that have higher prediction probabilities than τ_s are then saved and returned as the frequent itemsets.

Table 9: Rule quality evaluation of TFMs and baselines on small tabular datasets. In small tabular data, algorithmic ARM (FP-G) still leads to rule explosion. TFMs, XGBoost, and Aerial+ address rule explosion while TFMs and XGBoost find rules of higher statistical significance (Zhang and Interestingness). FP-G (x) refers to FP-Growth with minimum support threshold x.

Algorithm	**#Rules**	**Support**	**Confidence**	**Zhang**	**Interestingness**	**Coverage**
			Acute Inflammations			
TabICLv2	60.4	31.1	78.2	56.6	45.0	1
TabPFNv3	56.7	31.7	79.5	57.7	46.1	1
TabDPT	52.6	32.4	80.8	60.7	47.5	1
Aerial+	138.2	25.1	87.7	46.3	39.4	1
XGBoost	16.0	25.9	86.4	59.7	34.2	1
FP-G (0.5)	1	50.8	100.0	49.2	66.7	50.8
FP-G (0.3)	46	37.7	92.4	62.1	58.8	1
FP-G (0.2)	75	32.7	95.3	60.6	53.7	1
FP-G (0.1)	123	25.6	96.1	55.1	42.6	1
			Hepatitis			
TabICLv2	673.4	42.5	82.7	17.2	44.5	1
TabPFNv3	545.7	44.3	83.7	18.4	46.3	1
TabDPT	568.5	42.1	82.1	15.2	43.9	1
Aerial+	677.0	37.1	87.2	16.4	40.9	1
XGBoost	513.0	41.6	82.2	17.1	43.9	1
FP-G (0.5)	304	58.9	87.4	21.7	62.4	1
FP-G (0.3)	835	45.9	88.7	19.4	49.7	1
FP-G (0.2)	1153	40.2	88.9	18.0	43.8	1
FP-G (0.1)	2340	26.5	89.9	18.3	29.6	1
			Fertility			
TabICLv2	47.5	36.3	85.8	11.1	36.7	1
TabPFNv3	42.8	39.8	87.0	2.1	39.6	1
TabDPT	42.2	39.7	87.7	7.5	39.8	1
Aerial+	73.4	31.6	86.7	8.5	33.1	1
XGBoost	279.0	20.5	67.9	12.0	21.3	1
FP-G (0.5)	10	59.6	88.9	7.9	60.1	67.1
FP-G (0.3)	52	41.8	88.5	3.6	42.1	1
FP-G (0.2)	109	32.8	88.5	5.9	33.7	1
FP-G (0.1)	287	20.5	89.3	10.3	22.0	1
			Cervical Cancer Behavior Risk			
TabICLv2	10.4	33.3	82.7	31.2	36.7	87.3
TabPFNv3	11.2	43.1	94.0	56.4	46.5	89.2
TabDPT	37.6	29.6	80.7	38.4	33.2	98.5
Aerial+	24.1	29.6	86.2	28.3	35.4	92.9
XGBoost	9.0	41.5	90.4	43.8	44.7	98.6
FP-G (0.5)	2	61.1	96.8	22.8	65.9	63.2
FP-G (0.3)	27	40.4	90.7	18.4	43.9	98.6
FP-G (0.2)	83	29.2	92.0	24.3	34.3	1
FP-G (0.1)	476	16.6	92.7	31.6	23.8	1
			Autism Screening Adolescent			
TabICLv2	861.6	38.9	84.9	20.2	40.9	1
TabPFNv3	713.6	43.9	88.1	17.5	45.6	1
TabDPT	773.2	39.0	83.1	9.1	40.1	1
Aerial+	663.7	33.8	89.9	14.2	37.0	1
XGBoost	645.0	47.4	89.6	31.1	49.4	1
FP-G (0.5)	445	58.5	89.0	17.6	61.0	1
FP-G (0.3)	1193	46.3	88.9	14.8	48.8	1
FP-G (0.2)	1982	37.5	89.5	13.5	39.8	1
FP-G (0.1)	4088	25.5	90.2	14.8	28.0	1

Table 10: Rule quality evaluation of TFMs and baselines on larger tabular datasets. TFMs reach full data coverage with up to 12 times fewer rules than FP-Growth (Spambase), addressing rule explosion, while achieving higher statistical significance (Zhang and Interestingness). FP-G (x) refers to FP-Growth with minimum support threshold x.

Algorithm	**#Rules**	**Support**	**Confidence**	**Zhang**	**Interestingness**	**Coverage**
			Breast Cancer			
TabICLv2	35.1	45.1	86.7	52.3	52.3	96.3
TabPFNv3	26.8	48.5	88.5	59.9	57.0	92.3
TabDPT	27.0	47.5	88.2	58.5	55.8	91.7
Aerial+	15.7	43.9	88.1	36.4	52.4	90.2
XGBoost	1.0	61.9	88.1	56.5	71.3	70.3
FP-G (0.5)	22	59.5	87.6	46.5	68.7	94.4
FP-G (0.3)	51	44.4	87.8	32.1	51.4	94.4
FP-G (0.2)	112	33.4	87.7	26.7	39.2	97.9
FP-G (0.1)	208	24.5	87.3	23.4	29.1	100.0
			Congressional Voting Records			
TabICLv2	565.8	33.4	88.5	83.2	56.0	99.8
TabPFNv3	423.4	34.6	90.1	85.8	58.1	99.8
TabDPT	312.6	34.5	89.4	83.1	56.3	99.8
Aerial+	224.0	34.8	92.1	61.4	56.6	97.8
XGBoost	11.0	28.9	92.8	94.5	67.0	75.9
FP-G (0.5)	14	51.6	89.8	84.7	78.4	57.6
FP-G (0.3)	1016	35.5	89.1	70.6	60.6	100.0
FP-G (0.2)	2024	30.7	88.6	65.0	52.9	100.0
FP-G (0.1)	2867	26.0	88.5	60.5	44.7	100.0
			Mushroom			
TabICLv2	1912.6	29.0	73.9	21.6	32.6	100.0
TabPFNv3	1715.4	29.3	76.4	25.2	34.0	100.0
TabDPT	1752.2	27.1	67.3	11.8	30.3	100.0
Aerial+	857.6	37.2	95.5	8.6	39.4	100.0
XGBoost	4326.0	19.8	62.0	8.7	21.4	100.0
FP-G (0.5)	245	65.8	94.5	6.8	66.0	100.0
FP-G (0.3)	1247	44.3	93.7	17.8	48.6	100.0
FP-G (0.2)	3147	31.8	95.0	24.5	38.7	100.0
FP-G (0.1)	6882	22.2	95.3	27.6	29.3	100.0
			Chess (KR vs. KP)			
TabICLv2	13710.0	56.4	90.7	10.0	57.2	100.0
TabPFNv3	12988.7	57.6	91.9	10.8	58.4	100.0
TabDPT	14454.4	56.4	90.0	8.7	57.0	100.0
Aerial+	11901.7	57.4	94.8	1.3	57.6	100.0
XGBoost	11688.0	49.3	84.4	8.2	49.3	100.0
FP-G (0.5)	12752	69.7	93.0	4.9	69.0	100.0
FP-G (0.3)	21064	57.4	93.1	4.5	57.2	100.0
FP-G (0.2)	27604	49.8	93.3	4.8	49.9	100.0
FP-G (0.1)	35358	42.1	93.5	5.6	42.6	100.0
			Spambase			
TabICLv2	46180.4	77.3	94.8	5.2	77.7	100.0
TabPFNv3	44268.8	78.3	94.8	5.1	78.7	100.0
TabDPT	9340.8	63.6	90.1	20.0	65.2	100.0
Aerial+	23075.1	81.4	97.7	2.5	81.7	100.0
XGBoost	27530.0	57.1	90.9	2.8	57.5	100.0
FP-G (0.5)	96094	77.8	92.2	4.2	76.5	100.0
FP-G (0.3)	109312	73.6	92.2	4.7	72.6	100.0
FP-G (0.2)	113168	72.0	92.3	4.9	71.2	100.0
FP-G (0.1)	131567	63.4	93.2	4.9	62.6	100.0

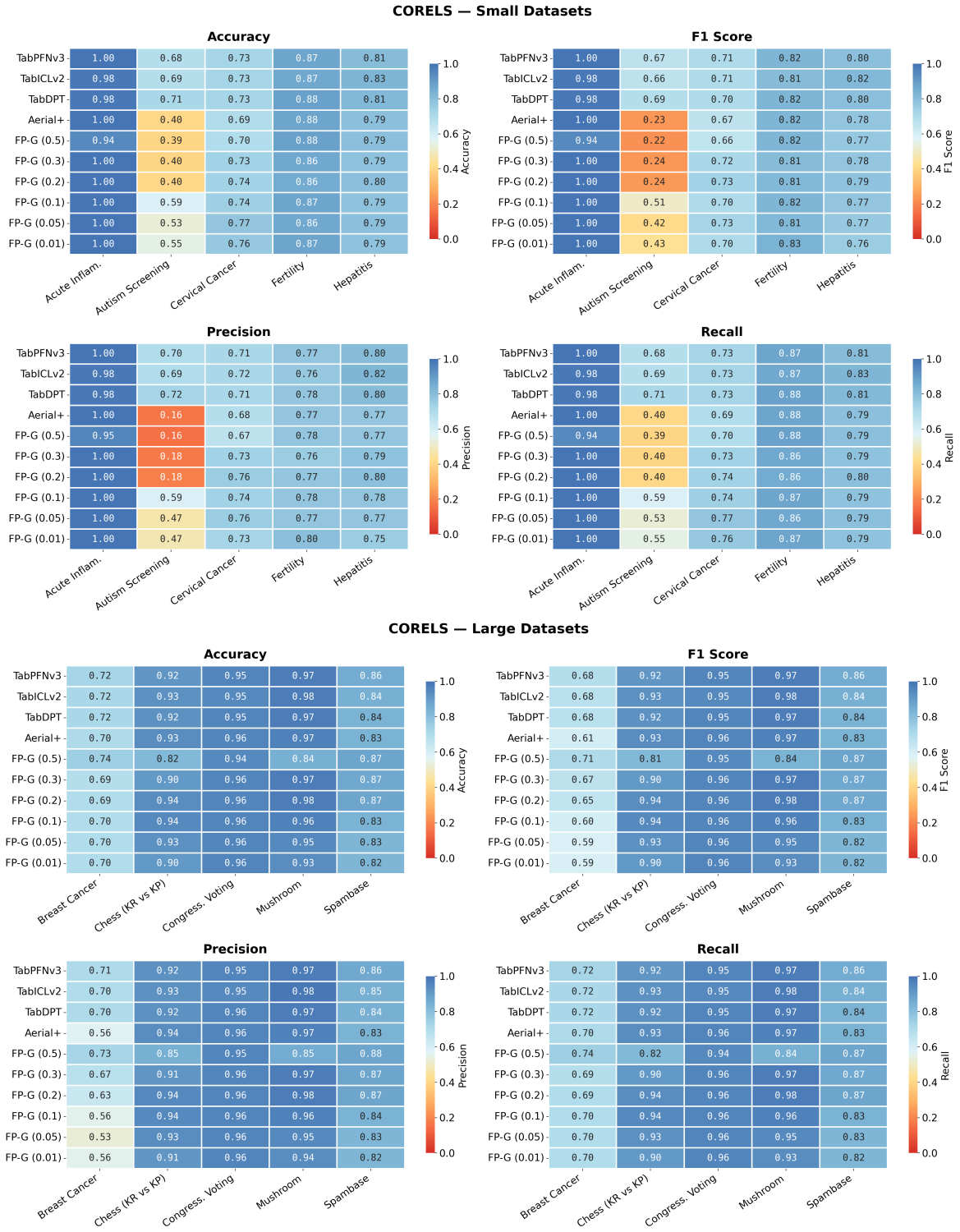

Figure 5: Classification results with CORELS on small (top) and larger (bottom) tabular datasets. On small datasets, TFMs have better or comparable ($\pm 2\%$) performance across all datasets, while Aerial+ falls behind FP-Growth on Autism Screening and Cervical Cancer. On larger datasets, TFMs again achieve better or comparable performance, with Aerial+ recovering but remaining $\sim 1\%$ behind TabICLv2 and TabPFNv3.

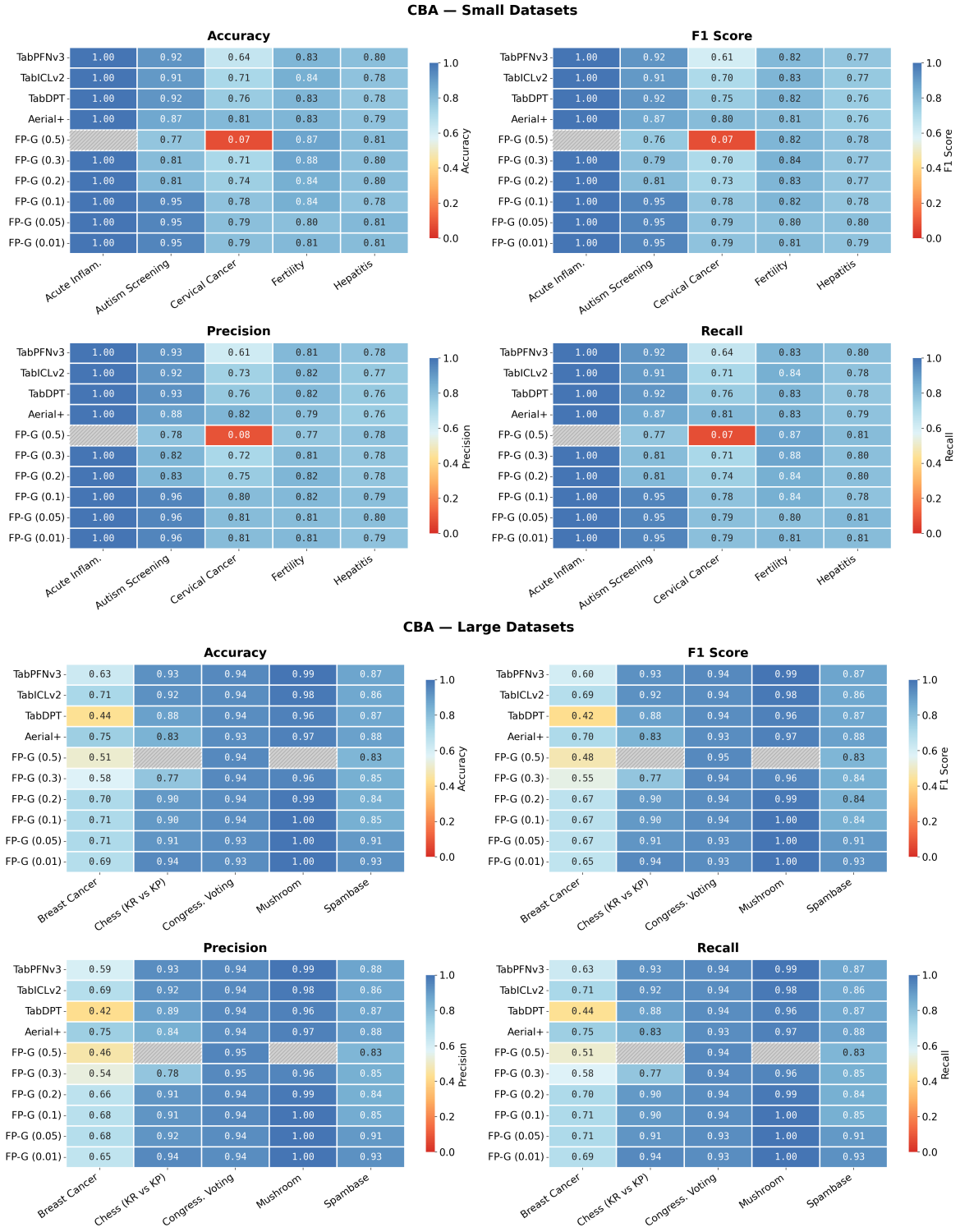

Figure 6: Classification results with CBA on small (top) and larger (bottom) tabular datasets. TFMs have better or comparable performance in 3 out of 5 small datasets and 4 out of 5 larger datasets. Note that CBA prioritizes rules of higher support and confidence rather than association strength and generalizability.

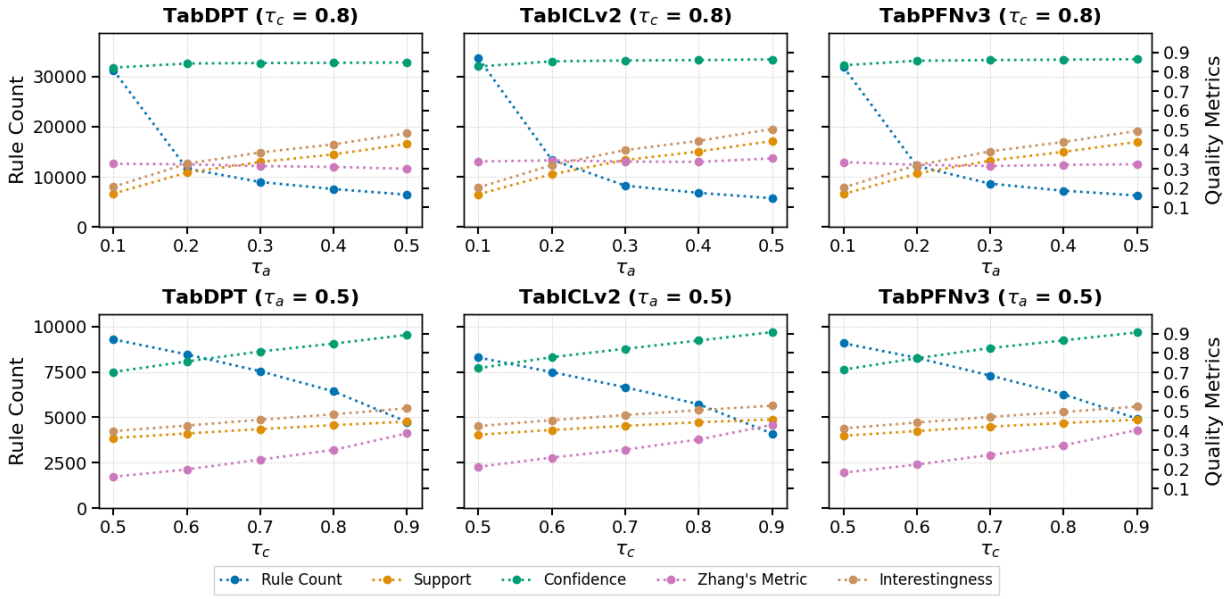

Figure 7: Hyperparameter analysis of Algorithm 2 over all 10 datasets. The left y-axis specifies rule counts, and the right y-axis specifies rule quality scores. Increasing τ_a (while keeping $\tau_c = 0.8$ fixed) leads to fewer rules with higher support and interestingness, while association strength (Zhang) and confidence do not change significantly. Increasing τ_c (while keeping $\tau_a = 0.5$ fixed) leads to higher support, confidence, association strength, and interestingness while reducing rule count.