PAVES: PRIOR-GUIDED DATASET COMPILATION FOR EFFECTIVE TRAINING AND SCALABLE EXPANSION

Anonymous authors
Paper under double-blind review

ABSTRACT

High-quality tabular datasets are often too small for effective model training because representative records are costly to collect. Synthetic expansion can increase the training size, but finite observations may contain spurious cross-column relationships. Sampling more records from a fixed generated distribution reduces variance without removing this relation bias, causing downstream models to learn misspecified structure more consistently. We introduce PAVES (**P**rior-guided **A**cquisition and **V**alidation of **E**xecutable **S**pecifications), a framework for prior-guided relation-space compression and executable dataset generation. Prior knowledge narrows the candidate relationships, while observed data determine which candidates are retained and estimate their parameters. Under coverage and separation conditions, compression reduces the uniform finite-sample error of relation validation; we further connect the remaining relation error to task-relevant distribution shift. PAVES compiles validated relationships and statistically estimated marginals into an editable executable specification that can be reused to replay a dataset at its source size or generate larger training sets without retraining an implicit generator. Across tabular benchmarks, PAVES achieves strong statistical and specification fidelity and downstream utility, while target-size experiments show continued gains when the same specification is reused at larger scales. Code is provided in the supplementary material.

1 INTRODUCTION

High-quality tabular datasets support a wide range of machine-learning applications, but collecting enough representative records is often expensive. A tabular distribution consists not only of the marginal distribution of each column, but also of the dependencies that determine how columns vary together. Synthetic-data methods provide a practical way to replay a finite dataset at its original size or expand it into a larger training set. Prominent approaches learn an implicit generator using adversarial training, variational inference, diffusion, or autoregressive modeling (Xu et al., 2019; Kotelnikov et al., 2023; Solatorio & Dupriez, 2023; Zhang et al., 2024). Although these models can sample an arbitrary number of rows, generating more records is useful for downstream learning only when the generated data preserve the relational features and distributional features that matter to prediction.

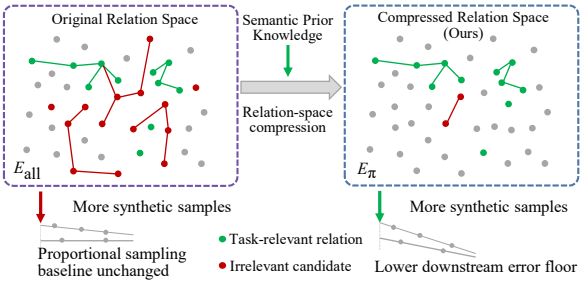


Figure 1: Prior-guided compression restricts validation from E_{all} to E_{π} and reduces finite-sample relation-selection error.

Column marginals can be estimated with standard statistics, whereas relation discovery is more difficult: in wide tables, the number of type-compatible pairwise, conditional, and higher-order candidates grows rapidly, making exhaustive validation complex. End-to-end generators fitted to finite samples can therefore encode spurious empirical correlations. Increasing the target size of a fixed generated distribution reduces sampling variability around, rather than removes, this biased rela-

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.

tion structure; downstream models may learn misspecified relationships more consistently, leaving relation bias as an error floor, as illustrated in Figure 1.

Our key observation is that semantic prior knowledge should compress the set of relationships to be measured, while observed data should retain final authority over acceptance and parameter estimation. Let E_{all} denote the complete type-compatible candidate set, $E^\star$ the task-relevant relations, and E_π the prior-guided candidate set. Under the coverage condition $E^\star \subseteq E_\pi \subsetneq E_{\text{all}}$, the prior removes or downweights implausible candidates while preserving task-relevant relations. The uniform validation error then depends on $\log |E_\pi|$ rather than $\log |E_{\text{all}}|$. With a sufficient separation margin between true and false relations, the smaller error supports recovery of the relevant relation graph and suppresses spurious dependencies caused by finite observations.

This relation-space result also explains when dataset expansion can improve downstream learning. Relation-estimation error induces a task-relevant shift between the source distribution and the generated distribution. A downstream risk bound contains both this shift term and a finite-training-sample term. Increasing the generated training-set size reduces the latter, but not the former. By lowering the relation error before generation, prior-guided compression reduces the shift term and permits the model trained on replayed or expanded data to benefit from a larger effective training set. Thus, our objective is not merely to produce additional rows, but to compile and generate finite observations in a form that yields better downstream generalization under the stated coverage, separation, and implementation conditions.

We introduce PAVES (*Prior-guided Acquisition and Validation of Executable Specifications*) as an executable realization of this theoretical framework. PAVES uses a large language model to map prior knowledge from the schema and application context to a typed candidate set E_π. A data-driven constraint extractor evaluates each candidate on the source data, estimates its relation-specific parameters, and retains a validated relation set $\widehat{E}$. Column marginals are estimated statistically and shared across the compared relation-selection procedures. The retained marginals and relationships are consolidated into an editable explicit representation file, which can be compiled into allocation, sampling, feasibility, and procedural operators and executed to produce a dataset of target size N.

The specification replays the source dataset structure as a new training table and can also expand the dataset from the same specification without retraining an implicit generator. Models are trained on the resulting data and evaluated on held-out real observations. Statistical fidelity and relation-recovery measurements diagnose the mechanism, while downstream utility and target-size scaling evaluate its consequence. In summary, our contributions are:

- We connect relation-estimation error to task-relevant distribution shift and downstream generalization, explaining why a lower relation error yields a lower risk bound at the same generated training-set size and supports larger dataset expansion under a fixed shift budget.
- We present PAVES, which implements the framework by translating prior knowledge into typed candidate relations, validating and quantifying them from data, and compiling the resulting marginals and relationships into an editable executable specification for dataset replay and expansion.
- We provide a systematic empirical evaluation of relation recovery, statistical fidelity, downstream utility, and target-size scaling, demonstrating that the same specification can generate effective training data across increasing target sizes.

2 Related Work

Synthetic data generation. PrivBayes (Zhang et al., 2017) approximates high-dimensional joint distributions using Bayesian networks and low-dimensional marginals. CTGAN (Xu et al., 2019) uses conditional adversarial training and mode-specific normalization to generate mixed-type tabular data with imbalanced categorical variables. TabDDPM (Kotelnikov et al., 2023) applies Gaussian and multinomial diffusion processes to continuous and categorical table columns. CoDi (Lee et al., 2023) couples separate diffusion models for continuous and discrete variables through contrastive conditioning. GReaT (Borisov et al., 2022) serializes rows and fine-tunes an autoregressive language model for conditional tabular generation. TabMT (Gulati & Roysdon, 2023) uses masked-token modeling to generate heterogeneous table rows with missing values. TabSyn (Zhang et al.,

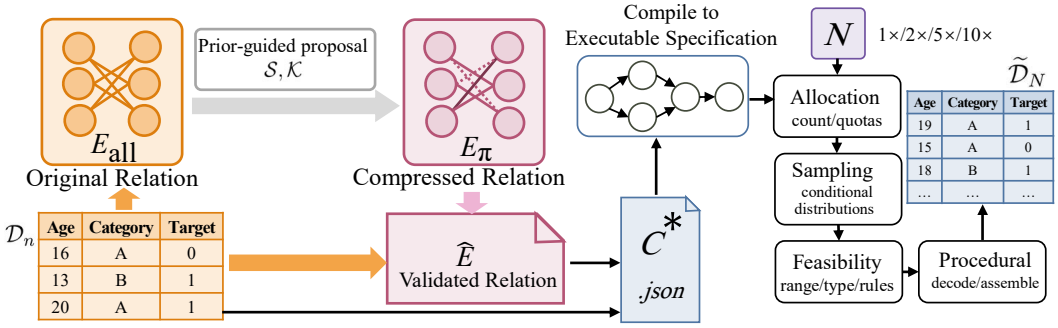
2024) performs score-based diffusion in a shared latent space learned for mixed-type tabular data. Structure-aware generators explicitly model cross-column dependence: GOGGLE (Liu et al., 2023) jointly learns a relational graph and the corresponding generative mechanisms, SPADA (Yang et al., 2025) uses an LLM-induced sparse dependency graph to guide nonparametric or flow-based synthesis, whereas LLM-TabLogic (Long et al., 2025) injects LLM-extracted logical relationships into latent diffusion. Constraint-conditioned methods incorporate additional generation requirements: KAMINO (Ge et al., 2020) preserves integrity constraints under differential privacy, CuTS (Vero et al.) translates supplied statistical and logical constraints into differentiable objectives, and C-DGMs (Stoian et al., 2024) enforce supplied numerical relations through a constraint layer.

Relation and constraint discovery. HyFD (Papenbrock & Naumann, 2016) combines sampling-based candidate discovery with full-data validation to extract minimal functional dependencies. FDX (Zhang et al., 2020) formulates functional-dependency discovery as a sparse regression problem. FASTDC (Chu et al., 2013) derives denial constraints from predicate evidence and minimal-cover search over relational instances. Sparse graphical-model discovery learns a restricted dependency structure from empirical statistics and illustrates how structural priors can aid relation recovery (Belilovsky et al., 2017). Sherlock (Hulsebos et al., 2019) infers column semantic types from statistical, character-level, and embedding-based features. Sato (Zhang et al., 2019) incorporates table context and structured prediction into semantic column-type detection. Cocoon (Huang & Wu, 2024) combines LLM-derived semantic context with statistical profiling to identify semantically meaningful data properties. Auto-Test (Chen et al., 2025) learns reusable semantic-domain constraints from large table corpora through statistical testing and constraint selection. Guardrail (Ma et al., 2025) synthesizes executable integrity-constraint programs from noisy data.

Positioning. PAVES formalizes semantic guidance as relation-hypothesis-space compression: the semantic model proposes a typed candidate set, while data-driven extraction determines which candidates are supported and estimates their parameters. This differs from trusting an LLM-induced graph as the generation structure, injecting semantic relationships into a learned generator, or repairing another generator's outputs. The validated relationships and shared marginal estimates are consolidated into an editable specification and compiled into a complete replay and expansion process.

3 METHODOLOGY

We first formulate the relation-selection problem and then present PAVES as a complete procedure for proposing, extracting, compiling, and executing dataset relationships. After defining the resulting generation process, we analyze why prior-guided relation-space compression improves finite-sample relation recovery and how this improvement affects downstream risk. Figure 2 summarizes the four implementation stages.



Age	Category	Target
16	A	0
13	B	1
20	A	1

Age	Category	Target
19	A	1
15	A	0
18	B	1
…	…	…

Figure 2: Overview of PAVES as an executable realization of prior-guided relation-space compression. Semantic prior knowledge maps the full candidate space E_{all} to a smaller measurement set E_π; data-grounded extraction validates and quantifies the candidates to obtain $\widehat{E}$. The retained marginals and relations are compiled into an execution graph and materialized as a replayed or expanded training dataset of size N.

3.1 PROBLEM FORMULATION

Let the observed table be $\mathcal{D}_n = \{\mathbf{z}_i\}_{i=1}^n$, where $\mathbf{z}_i \overset{\text{i.i.d.}}{\sim} P$ denotes a row independently drawn from the population distribution P, and $\mathbf{z}_i \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$ specifies its product space over the p column domains. Individual-column marginals are estimated identically for all compared relation-selection procedures; our analysis focuses on cross-column relations. Let E_{all} be all type-compatible candidates and associate each e with a bounded, positively oriented statistic $\phi_e(\mathbf{z}) \in [0, 1]$:

$$r_e(P) = \mathbb{E}_P[\phi_e(\mathbf{z})], \qquad \widehat{r}_e = \frac{1}{n}\sum_{i=1}^{n} \phi_e(\mathbf{z}_i). \tag{1}$$

Let $E^\star$ denote the population relations targeted for retention, with the separation assumption below distinguishing them from irrelevant candidates. The semantic prior returns E_π under the coverage condition $E^\star \subseteq E_\pi \subsetneq E_{\text{all}}$. Hence its graph class contains $2^{|E_\pi|}$ structures rather than $2^{|E_{\text{all}}|}$. The prior only restricts which relations are measured; all acceptance decisions and numerical parameters remain data-derived.

3.2 PAVES: AN EXECUTABLE INSTANTIATION

Given schema $\mathcal{S}$, prior context $\mathcal{K}$, observations $\mathcal{D}_n$, and target size N, PAVES learns a specification $\mathcal{C}^*$ and executes it as

$$\mathcal{C}^* = \text{LearnSpecification}(\mathcal{D}_n, \mathcal{S}, \mathcal{K}), \qquad \widetilde{\mathcal{D}}_N = \text{Execute}(\text{Compile}(\mathcal{C}^*), N). \tag{2}$$

This realizes $E_{\text{all}} \to E_\pi \to \widehat{E} \to \mathcal{Q}(\widehat{E})$: semantic interpretation constructs E_π, extraction estimates $\widehat{E}$, and compilation materializes an admissible distribution. The shared marginals, validated relations, parameters, and hard requirements are stored as typed records $c = (s, \mathcal{V}, o, \theta, \nu, \xi, \rho, \eta)$, whose fields specify scope, involved variables, operator, data-estimated parameters, support, uncertainty, provenance, and execution semantics. Domain, Statistical, Relational, and Derived families cover individual-column properties, empirical targets, cross-column associations, and field derivations. The complete specification is stored in an editable, human-readable explicit representation file and is schema-checked before compilation; formal definitions are given in Appendix Sections A.1 and A.2 .

Prior-guided proposal. The semantic interpreter receives schema metadata, a data dictionary, domain context, the supported operator vocabulary, and a small preview used only to clarify value formats. It maps this prior knowledge to hypotheses $h = (s, \mathcal{V}, o, a, \rho, \gamma)$, where $\mathcal{V}$ identifies candidate variables, o is a supported extraction operation, a describes applicability, and (ρ, γ) record provenance and semantic confidence. After canonicalization, these hypotheses instantiate E_π; they cannot contain empirical parameters or executable programs.

Data-grounded extraction. Each hypothesis is quantified on $\mathcal{D}_n$ and retained only when it has sufficient support, corrected significance (or held-out fit), and effect. Validation returns $(\widehat{c}_h, \nu_h, q_h, e_h) = \text{Validate}(h, \mathcal{D}_n)$, and the retained set is $\mathcal{C}_{\text{val}} = \{\widehat{c}_h : \nu_h \geq m,\ q_h \leq \alpha,\ e_h \geq \epsilon\}$, where m, α, and ϵ are the minimum support, false-discovery, and type-specific effect thresholds. Depending on the operator, validation uses violation rate, residual error, association strength, or held-out likelihood gain.

Validated variables may define conditional profiles using support-limited splits and sparse backoff. Conflict-aware consolidation then removes redundant, over-specific, or incompatible measurements, prioritizing domain validity, validated deterministic relations, and statistical targets. Appendix Section A.3 provides the branching rule, consolidation objective, and conflict checks.

3.3 SPECIFICATION COMPILATION

Execution operators. The compiler maps every extracted constraint to one of four operator types. Allocation operators convert proportions and subgroup counts into integer quotas. Sampling operators draw from smoothed marginal or conditional distributions. Feasibility operators enforce

deterministic domain, ordering, uniqueness, and derivation requirements. Procedural operators materialize identifiers, text-like fields, and complex derived values under an explicitly registered procedure.

The compiler stores operator dependencies in a directed execution graph $G = (\mathcal{N}, \mathcal{E})$. An edge $n_i \to n_j$ means that n_j consumes an assignment or allocation produced by n_i. Mutually dependent variables are grouped into one joint component, and the graph between components is acyclic. For node v, generation follows $\mathbf{x}_v \sim g_v(\mathcal{C}_v^*, \{\mathbf{x}_u : u \in \mathrm{Pa}(v)\})$, depending only on its local constraints and direct parents. This local execution avoids repeatedly solving all ancestral predicates.

Constraint solving and dataset generation. Appendix Section A.4 provides implementation-level pseudocode for the complete generation procedure, including target stratification, quota allocation, batched feasibility checking, and table assembly.

Target-size scaling. For a target proportion p_g and dataset size N, an allocation operator uses largest-remainder rounding:

$$n_g = \lfloor N p_g \rfloor + \mathbb{I}[g \in \mathrm{TopResidual}(b_N)], \qquad b_N = N - \sum_g \lfloor N p_g \rfloor. \tag{3}$$

Therefore $\sum_g n_g = N$ and $|n_g/N - p_g| \le 1/N$. Exact subgroup counts are treated as proportions when N changes unless the specification marks them as fixed constants, and overlapping quotas are reconciled before row-level values are sampled.

Sampling operators use smoothed empirical conditionals and hierarchical backoff, rather than replaying observed rows or complete combinations. Randomized quota order, independent random seeds, uniqueness checks, and randomized feasible assignments reduce repeated outputs. If a component cannot be generated, the engine reports the minimal conflicting subset when available, removes no hard constraint silently, and either applies the operator's declared fallback or terminates with a failure record.

Correctness properties. The compilation contract gives two basic properties. First, if every compiled component completes and procedural operators preserve their declared postconditions, then $\widetilde{\mathcal{D}}_N \models \mathcal{C}_{\mathrm{hard}}^*$. This follows by induction over the execution order: each component satisfies its local hard constraints using assignments from already valid predecessor nodes, and row assembly preserves the validated domains and dependencies. Second, largest-remainder allocation contributes at most $1/N$ deterministic proportion error. For conditionally independent draws from the selected sampling target, the remaining empirical error has the usual finite-sample order

$$|\widehat{p}_g - p_g| \le \frac{1}{N} + O_p\left(\sqrt{\frac{p_g(1-p_g)}{N}}\right). \tag{4}$$

Our implementation uses Z3 as the feasibility solver.

3.4 Theoretical Analysis

Having defined how PAVES constructs E_π, validates $\widehat{E}$, and compiles the retained relations into a distribution Q, we now analyze the statistical principle underlying this design.

Relation recovery. Treat E_π and its relation scores as fixed before their empirical statistics are computed; formally, proposal information is independent of the validation sample, or relation validation uses a disjoint split. Suppose a threshold τ separates relevant and irrelevant candidates by a margin $\gamma > 0$. Hoeffding's inequality and a union bound give the following recovery guarantee:

$$n > \frac{\log(2|E_\pi|/\delta)}{2\gamma^2} \quad \Longrightarrow \quad \Pr(\widehat{E} = E^\star) \ge 1 - \delta, \tag{5}$$

where $\widehat{E} = \{e \in E_\pi : \widehat{r}_e \ge \tau\}$. Full candidate validation replaces $|E_\pi|$ with the larger $|E_{\mathrm{all}}|$ and therefore requires more observations for the same guarantee. The complete assumptions and proof are provided in Appendix Section A.1.

Downstream implication. Let Q be the compiled distribution and define its relation shift as $\Delta_{\mathrm{rel}}(P, Q) = \|r(Q) - r(P)\|_2$. We assume that this representation controls the task-relevant distribution difference up to uncaptured higher-order structure $\varepsilon_{\mathrm{high}}$. For a model trained on N idealized i.i.d. rows from Q, the resulting bound can be summarized as

$$R_P(\widehat{f}_{Q,N}) - \inf_{f \in \mathcal{F}} R_P(f) \leq 2L_{\mathrm{rel}}\Delta_{\mathrm{rel}}(P, Q) + 2\varepsilon_{\mathrm{high}} + \mathrm{Est}_{\mathcal{F}}(N), \tag{6}$$

where $\mathrm{Est}_{\mathcal{F}}(N)$ is the standard finite-training-sample term. Increasing N reduces this term but cannot remove relation shift. On the recovery event, prior-guided validation removes finite-sample perturbations along irrelevant relation coordinates; if compilation error does not offset this reduction, it yields a lower risk bound at the same N and supports expansion over a wider range of target sizes. Appendix Section A.1 gives the relation-error decomposition, the full Rademacher bound, and the expansion-ratio corollary.

4 Experiments

We evaluate PAVES through comparison with existing methods on standard tabular data, UMAP visualization of data, constraint-construction ablations, and scalable dataset expansion.

4.1 Experimental Details

Datasets. The standard tabular experiments use eleven real-world datasets: Faults, Wilt, HTRU2, Bean, Obesity, Titanic, Shoppers, and Indian Liver Patient are classification tasks, while News, Insurance, and Abalone are regression tasks. Dataset statistics are reported in Appendix Section B.

Comparison methods. For the standard tabular benchmarks, we compare PAVES with SMOTE (Chawla et al., 2002), CTGAN (Xu et al., 2019), TVAE (Xu et al., 2019), PATECTGAN (Jordon et al., 2018), TabDDPM (Kotelnikov et al., 2023), TabSyn (Zhang et al., 2024), GReaT (Borisov et al., 2022), SPADA (Yang et al., 2025), LLM-TabLogic (Long et al., 2025). Statistical fidelity is measured by Shape, Corr, Precision, and Recall. Downstream classification utility is evaluated with Decision Tree, Random Forest, XGBoost, and CatBoost.

Experiments settings. The large language model used to construct typed semantic hypotheses is GPT-5.6 Sol. All public datasets use their complete benchmark splits. The LLM, constraint extractor, and generator access only the training split; extraction hyperparameters are fixed on validation data before test generation. Unless the dataset-expansion experiment states otherwise, PAVES generates the same number of rows as the real training set and preserves the table schema. Downstream models are trained on NVIDIA H20 GPUs.

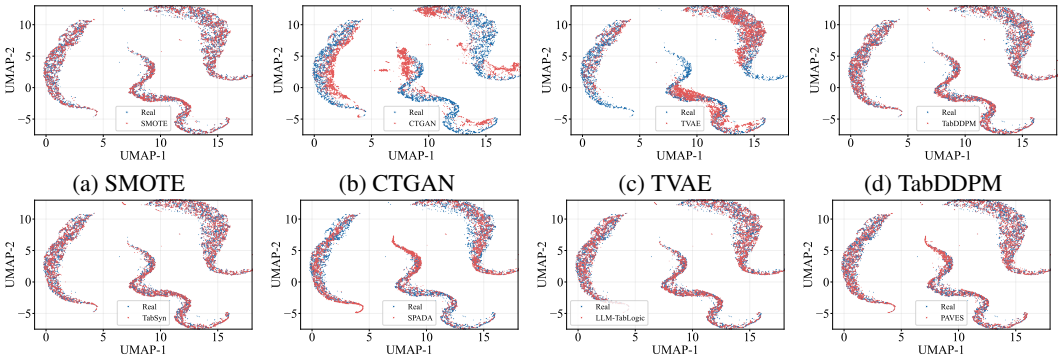
4.2 Standard Tabular Comparison

Standard classification and regression. Downstream prediction evaluates whether synthetic data support effective models, but it does not directly identify which properties of the source table are retained. We therefore complement TSTR with five direct measurements of generated-table quality. Precision and Recall evaluate sample-level fidelity and coverage following Alaa et al. (2022). Hard Constraint Satisfaction measures the fraction of retained deterministic requirements satisfied by the generated table. Conditional Distribution Fidelity measures agreement between the validated conditional targets and their empirical counterparts in the generated data. Statistical Target Error aggregates discrepancies between the generated table and the marginal, proportional, and numerical targets recorded in the executable specification. Together, these metrics evaluate both where generated records lie and whether execution preserves the properties represented by the specification.

Table 1 presents the averaged comparison. PAVES achieves the highest Precision (0.967), perfect Hard Constraint Satisfaction (1.000), and the highest Conditional Distribution Fidelity (0.990). It also obtains the lowest Statistical Target Error (0.001), substantially below the next-best value of 0.006 from SMOTE. SMOTE attains the highest Recall (0.790), while PAVES reaches 0.306, indicating that its primary advantage lies in the validity and precision of the generated sample space rather than maximal coverage. Detailed per-dataset results for PAVES are reported in Appendix Section C.1.

Table 1: Statistical and specification fidelity averaged across the eleven standard tabular benchmark datasets.

Metric	PATECTGAN	CTGAN	TVAE	TabSyn	LLM-TabLogic	SPADA	TabDDPM	SMOTE	PAVES (Ours)
Precision ↑	0.755	0.741	0.802	0.818	0.880	0.934	0.937	0.955	**0.967**
Recall ↑	0.079	0.120	0.130	0.346	0.374	0.555	0.427	**0.790**	0.306
Hard Constraint Satisfaction ↑	0.996	0.970	1.000	0.977	0.996	1.000	0.997	0.998	**1.000**
Conditional Distribution Fidelity ↑	0.813	0.905	0.891	0.982	0.969	0.963	0.963	0.983	**0.990**
Statistical Target Error ↓	0.040	0.042	0.030	0.007	0.019	0.012	0.026	0.006	**0.001**



(a) SMOTE (b) CTGAN (c) TVAE (d) TabDDPM

(e) TabSyn (f) SPADA (g) LLM-TabLogic (h) PAVES (Ours)

Figure 3: UMAP projections of real and synthetic samples for eight synthesis methods. Each pair of real and synthetic records is embedded jointly using the same fitted UMAP projection.

To measure downstream classification utility, we additionally conduct TSTR experiments on all eight classification tasks among the eleven standard tabular datasets. For each synthesis method, Decision Tree (Loh, 2011), Random Forest (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), and CatBoost (Prokhorenkova et al., 2018) classifiers are trained on the generated training set and evaluated on the corresponding untouched real test set. Table 2 reports the mean classification accuracy across the eight tasks for each downstream classifier. News, Insurance, and Abalone are excluded from this comparison because they are regression datasets. Detailed per-dataset results for PAVES are reported in Appendix Section C.2.

Table 2: Mean TSTR classification accuracy across the eight standard single-table classification datasets. Each row denotes a downstream classifier trained on synthetic data and evaluated on the corresponding real test set.

Method	Real	GReaT	CTGAN	TVAE	TabDDPM	TabSyn	SPADA	LLM-TabLogic	SMOTE	PAVES(Ours)
Decision Tree	0.820	0.530	0.586	0.743	0.780	0.799	0.738	0.780	0.837	**0.850**
Random Forest	0.859	0.621	0.644	0.756	0.808	0.861	0.780	0.850	0.860	**0.862**
XGBoost	0.857	0.598	0.636	0.763	0.819	**0.863**	0.785	0.854	**0.863**	**0.863**
CatBoost	0.860	0.611	0.644	0.764	0.832	0.864	0.787	0.855	0.859	**0.866**

Table 2 shows that datasets processed by PAVES yield competitive downstream performance. Among synthetic-data methods, PAVES obtains the highest mean accuracy with Decision Tree (0.850), Random Forest (0.862), and CatBoost (0.866), and ties TabSyn with XGBoost (0.863). Averaged over the four classifiers, PAVES reaches 0.860, compared with SPADA (0.773), LLM-TabLogic (0.835), and SMOTE (0.855). PAVES exceeds the real-data baseline with all four classifiers. These results indicate that the generated tables retain strong predictive utility under this evaluation.

4.3 Visualization of Real and Synthetic Data

We use UMAP to project real and synthetic records into the same two-dimensional space. Figure 3 compares PAVES with seven representative synthesis methods. The synthetic points produced by PAVES show substantial overlap with the regions occupied by the real data, indicating that the extracted specification preserves the main geometry and coverage of the observed sample space.

TabDDPM and TabSyn also achieve strong visual alignment with the real distribution. However, both methods require training high-capacity generative models and therefore incur substantial train-

ing computation, while changing a retained data property generally requires retraining. In contrast, PAVES extracts, compiles, and executes an explicit dataset specification without training an implicit generative model. This provides competitive distributional overlap while allowing individual constraints and statistical targets to be edited directly without retraining.

4.4 Scalable Dataset Expansion

This experiment evaluates how downstream performance changes when a fixed executable specification is reused at increasing target sizes. For each of the eight standard single-table classification datasets, PAVES extracts the specification once and reuses it without modification at $1\times$, $2\times$, $5\times$, and $10\times$ the source size. CTGAN, TVAE, SPADA, TabDDPM, LLM-TabLogic, TabSyn, and SMOTE are evaluated at the same multipliers. We additionally include a Real repetition baseline: its $1\times$ dataset is the original real training set, while its $2\times$, $5\times$, and $10\times$ datasets are formed by repeating the same real training records to the corresponding sizes. This control increases the number of training rows without introducing new observations or changing the empirical relation structure. A Transformer is trained on each dataset and evaluated on the same fixed real evaluation set. News, Insurance, and Abalone are excluded because they define regression tasks. Figure 4 presents four

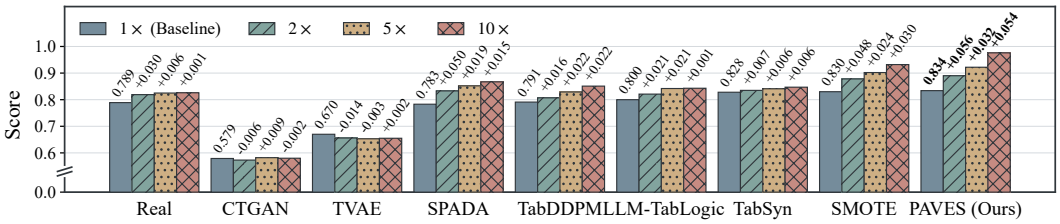


Figure 4: Mean Transformer Macro-F1 under increasing training-set sizes. For Real, $1\times$ is the original real training set, while $2\times$, $5\times$, and $10\times$ are constructed by repeating the same real training records to the corresponding sizes. Labels above the $1\times$ bars are absolute Macro-F1 scores. Each signed label at $2\times$, $5\times$, and $10\times$ denotes the change from the immediately preceding multiplier ($1\times\rightarrow2\times$, $2\times\rightarrow5\times$, and $5\times\rightarrow10\times$, respectively), rather than the change from the $1\times$ baseline.

groups of bars, one for each target-size multiplier. Each bar reports mean Transformer Macro-F1 across the eight classification datasets. For PAVES, every dataset-multiplier configuration is trained with seeds 42, 43, and 44. These seeds quantify downstream Transformer training variability; the synthetic dataset is fixed within each dataset-multiplier configuration. Model selection and early stopping use only a synthetic validation split, preprocessing statistics and categorical vocabularies are fitted only on the synthetic training split, and the fixed real evaluation set is accessed once after reloading the selected checkpoint. The labels above the $1\times$ bars report absolute Macro-F1. Each subsequent signed label reports the change from the immediately preceding multiplier rather than the cumulative change from $1\times$. For the Real baseline, these multipliers denote repeated copies of the original training records. Per-dataset, per-seed results for Accuracy, Macro-F1, weighted F1, and balanced accuracy are reported in Appendix Section C.3.

Across all four target sizes, PAVES achieves the highest mean Transformer Macro-F1 among the evaluated methods. Averaging each seed across the eight datasets, its Macro-F1 is 0.834, 0.890, 0.923, and 0.976 at $1\times$, $2\times$, $5\times$, and $10\times$, respectively, where the uncertainty is the sample standard deviation over the three downstream training seeds. The corresponding stepwise changes in the means are $+0.056$, $+0.032$, and $+0.054$. In comparison, repeating the real training rows improves the Real baseline by $+0.030$ at $2\times$, but the subsequent gains shrink to $+0.006$ and $+0.001$ at $5\times$ and $10\times$. Thus, under this protocol, increasing the number of presented training rows alone does not explain the sustained gains of PAVES. CTGAN and TVAE show small or negative changes at larger scales, while several other baselines improve more gradually. The per-dataset results are heterogeneous, so this experiment establishes the observed scaling behavior of the fixed specification rather than claiming that generated records contain information absent from the source data. It nevertheless shows that one compiled specification can be executed at multiple target sizes without retraining an implicit generator.

4.5 Ablation Studies

Targeted relation retention and shift. We evaluate whether the downstream gains of PAVES arise from targeted relation shifts rather than merely changing the number of modeled relations. No Relation Shift retains every measured relation without targeted adjustment. For a fair comparison, Random Relation Shift and Code-guided Relation Shift target the same number of relations as PAVES on each dataset and therefore leave the same number of relations retained. Random Relation Shift selects associations uniformly at random for adjustment, whereas Code-guided Relation Shift ranks them using predefined data-profiling and schema rules without semantic prior knowledge. Across the eight benchmark classification datasets, the three shift variants adjust an average of 80.25 relations. All variants use the same marginal estimates, target size, specification compiler, generation procedure, and downstream evaluation protocol.

Table 3 reports the effects of relation adjustment or selection and the resulting downstream performance. Attenuation measures the normalized decrease in relation strength over the relations selected for adjustment: a positive value indicates that the targeted relations are weakened, whereas a negative value indicates that their measured strength increases after generation. Retention Error measures the discrepancy between real and generated relation strengths over relations retained in the specification; a lower value indicates that relation selection has a more localized effect and causes less disturbance to the retained structure. The four downstream columns report mean TSTR accuracy across the eight classification datasets for Decision Tree, Random Forest, XGBoost, and CatBoost.

Table 3: Ablation of targeted relation retention and shift. Comparison of Random Relation Shift, Code-guided Relation Shift, and PAVES across datasets. All four downstream metrics report the mean TSTR accuracy over the eight classification datasets.

Method	Shifted Relations	Attenuation	Retention Error	Decision Tree	Random Forest	XGBoost	CatBoost
No Relation Shift	0	0	0.025	0.848	0.859	0.855	0.864
Random Relation Shift	80.25	-1.239	0.122	0.811	0.843	0.844	0.842
Code-guided Relation Shift	80.25	-0.924	0.076	0.843	0.858	0.862	0.861
PAVES (Prior-guided Relation Shift)	80.25	0.635	0.038	0.850	0.862	0.863	0.866

PAVES is the only relation-shift variant with positive Attenuation (0.635), indicating that the relations targeted for adjustment are weakened on average. Random Relation Shift and Code-guided Relation Shift yield negative scores, indicating that their targeted relations become stronger on average after generation. A possible explanation is that changing which relations are retained does not enforce independence: retained dependencies can indirectly reproduce or strengthen a targeted association during sampling. These results highlight the importance of which relations are retained and shifted. By combining semantic guidance with data-grounded validation, PAVES achieves positive attenuation while limiting changes to the retained structure. No Relation Shift, which retains all measured relations without targeted adjustment, has the lowest observed Retention Error (0.025). Among the three variants with the same relation-shift budget, PAVES achieves the lowest Retention Error (0.038, compared with 0.076 for Code-guided Relation Shift and 0.122 for Random Relation Shift) and the highest accuracy across all four downstream classifiers. Together, these results support the view that the downstream benefit depends on the choice of retained and shifted relations, beyond the number of modeled relations alone.

5 Conclusion

We presented PAVES, a framework for prior-guided relation-space compression and executable tabular data generation. Our theoretical analysis shows that, when the compressed candidate space fully preserves task-relevant relations and these relations are statistically separable from irrelevant candidates, compression reduces finite-sample relation-selection error; the remaining relation error induces task-relevant distribution shift and affects downstream model risk. PAVES constructs candidate relations from semantic prior knowledge, validates them and estimates their parameters from observed data, and integrates the resulting relation structure and marginal distributions into an editable and compilable explicit specification. Without retraining an implicit generator, this specification can replay the source data distribution or produce larger training sets. Across multiple tabular benchmark datasets, PAVES performs strongly in all evaluations, demonstrating that a unified specification can transform limited source data into high-quality training sets and improve downstream model performance.

AI USE STATEMENT

In this work, we used generative AI tools for checking wording, typos and grammar during manuscript writing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We prioritize the reproducibility of this work. The source code of the PAVES framework, experiment scripts, and plotting code are provided in the supplementary materials. Full details regarding evaluation metrics, random seeds, model configurations, and experimental data can be found in the main text and appendix.

REFERENCES

Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International conference on machine learning*, pp. 290–306. PMLR, 2022.

Eugene Belilovsky, Kyle Kastner, Gaël Varoquaux, and Matthew B Blaschko. Learning to discover sparse graphical models. In *International conference on machine learning*, pp. 440–448. PMLR, 2017.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*, 2022.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Qixu Chen, Yeye He, Raymond Chi-Wing Wong, Weiwei Cui, Song Ge, Haidong Zhang, Dongmei Zhang, and Surajit Chaudhuri. Auto-test: Learning semantic-domain constraints for unsupervised error detection in tables. *Proceedings of the ACM on Management of Data*, 3(3):1–27, 2025.

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.

Xu Chu, Ihab F Ilyas, and Paolo Papotti. Discovering denial constraints. *Proc. VLDB Endow.*, 6 (13):1498–1509, 2013.

Chang Ge, Shubhankar Mohapatra, Xi He, and Ihab F Ilyas. Kamino: Constraint-aware differentially private data synthesis. *arXiv preprint arXiv:2012.15713*, 2020.

Manbir Gulati and Paul Roysdon. Tabmt: Generating tabular data with masked transformers. *Advances in Neural Information Processing Systems*, 36:46245–46254, 2023.

Zezhou Huang and Eugene Wu. Cocoon: Semantic table profiling using large language models. In *Proceedings of the 2024 workshop on human-in-the-loop data analytics*, pp. 1–7, 2024.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on knowledge discovery & data mining*, pp. 1500–1508, 2019.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pp. 17564–17579. PMLR, 2023.

Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pp. 18940–18956. PMLR, 2023.

Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. Goggle: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2023.

Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

Yunbo Long, Liming Xu, and Alexandra Brintrup. Llm-tablogic: preserving inter-column logical relationships in synthetic tabular data via prompt-guided latent diffusion. *arXiv preprint arXiv:2503.02161*, 2025.

Pingchuan Ma, Zhaoyu Wang, Zhenlan Ji, Zongjie Li, Ao Sun, and Shuai Wang. Guardrail: Automated integrity constraint synthesis from noisy data. *Proceedings of the ACM on Management of Data*, 3(4):1–26, 2025.

Thorsten Papenbrock and Felix Naumann. A hybrid approach to functional dependency discovery. In *proceedings of the 2016 International Conference on Management of Data*, pp. 821–833, 2016.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Aivin V Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041*, 2023.

Mihaela CÄ Stoian, Salijona Dyrmishi, Maxime Cordy, Thomas Lukasiewicz, and Eleonora Giunchiglia. How realistic is your synthetic data? constraining deep generative models for tabular data. *arXiv preprint arXiv:2402.04823*, 2024.

M Vero, M Balunović, and M Vechev. Cuts: customizable tabular synthetic data generation. 02 jun 2024. arxiv. *arXiv preprint arXiv:2307.03577*.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

Shuo Yang, Zheyu Zhang, Bardh Prenkaj, and Gjergji Kasneci. Doubling your data in minutes: Ultra-fast tabular data generation via llm-induced dependency graphs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10348–10369, 2025.

Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. Sato: Contextual semantic type detection in tables. *arXiv preprint arXiv:1911.06311*, 2019.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *International Conference on Learning Representations*, volume 2024, pp. 52829–52857, 2024.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

Yunjia Zhang, Zhihan Guo, and Theodoros Rekatsinas. A statistical perspective on discovering functional dependencies in noisy data. In *Proceedings of the 2020 acm sigmod international conference on management of data*, pp. 861–876, 2020.

A METHOD DETAILS

A.1 RELATION-RECOVERY AND DOWNSTREAM-RISK DETAILS

This section states the assumptions under which the theoretical claims in the main paper hold and supplies their proofs. Individual-column marginals are estimated identically for all relation-selection procedures and collected as $\widehat{\mathcal{M}}_n = (\widehat{P}_{1,n}, \ldots, \widehat{P}_{p,n})$; the comparison therefore isolates cross-column relation selection.

Scope and assumptions. Let $\mathcal{D}_n = \{\mathbf{z}_i\}_{i=1}^n$ be an i.i.d. validation sample from P. The following conditions are required.

- (A1) The proposal information used to construct E_π is independent of $\mathcal{D}_n$. Equivalently, all candidates and score definitions are fixed conditional on schema and prior context, or a disjoint proposal split is used. If candidate construction reuses the validation rows, the bound below must instead be uniform over the data-dependent proposal class, and replacing $|E_{\mathrm{all}}|$ by $|E_\pi|$ is not automatically valid.
- (A2) Each candidate e has a fixed positively oriented score $\phi_e(\mathbf{z}) \in [0,1]$, with $r_e(P) = \mathbb{E}_P[\phi_e(\mathbf{z})]$ and $\widehat{r}_e = n^{-1}\sum_{i=1}^n \phi_e(\mathbf{z}_i)$. Operators based on pairwise statistics, learned binning, conditional mutual information, or fitted likelihoods require either sample splitting or their own concentration bound; the Hoeffding result below applies directly only to bounded row-decomposable scores.
- (A3) The prior satisfies coverage and strict compression: $E^\star \subseteq E_\pi \subsetneq E_{\mathrm{all}}$, where $E^\star$ is the set of population relations targeted for retention.
- (A4) There are $\tau \in (0,1)$ and $0 < \gamma < \min\{\tau, 1-\tau\}$ such that $r_e(P) \geq \tau + \gamma$ for $e \in E^\star$ and $r_e(P) \leq \tau - \gamma$ for $e \in E_\pi \setminus E^\star$. Thus, the result assumes a beta-min separation and contains no candidate in the unresolved interval $(\tau - \gamma, \tau + \gamma)$.

These assumptions make explicit that semantic guidance restricts the measurement set but does not determine acceptance or empirical parameters.

Hypothesis-space compression. Let $M_\pi = |E_\pi|$ and $M_{\mathrm{all}} = |E_{\mathrm{all}}|$. If each candidate relation may be retained or rejected, the corresponding graph classes have cardinalities $|\mathcal{G}_\pi| = 2^{M_\pi}$ and $|\mathcal{G}_{\mathrm{all}}| = 2^{M_{\mathrm{all}}}$. Therefore

$$\log|\mathcal{G}_\pi| = M_\pi \log 2 < M_{\mathrm{all}} \log 2 = \log|\mathcal{G}_{\mathrm{all}}|. \tag{7}$$

This statement concerns structural selection only. Any continuous function fitted inside an accepted relation contributes additional complexity that is not removed by counting graph structures.

Proposition 1: uniform validation. Under (A1)–(A2), define

$$\epsilon_{n,\pi} = \sqrt{\frac{\log(2M_\pi/\delta)}{2n}}. \tag{8}$$

Then, with probability at least $1-\delta$,

$$\sup_{e \in E_\pi} |\widehat{r}_e - r_e(P)| \leq \epsilon_{n,\pi}. \tag{9}$$

For a fixed e, Hoeffding's inequality gives $\Pr(|\widehat{r}_e - r_e(P)| > t) \leq 2\exp(-2nt^2)$. A union bound over the M_π fixed candidates gives $2M_\pi \exp(-2nt^2)$; setting this quantity to δ proves the claim. Repeating the argument over all candidates gives $\epsilon_{n,\mathrm{all}} = \sqrt{\log(2M_{\mathrm{all}}/\delta)/(2n)}$. Hence $M_\pi < M_{\mathrm{all}}$ implies $\epsilon_{n,\pi} < \epsilon_{n,\mathrm{all}}$.

Corollary 1: exact recovery. Let $\widehat{E} = \{e \in E_\pi : \widehat{r}_e \geq \tau\}$. Under (A3)–(A4), if $\epsilon_{n,\pi} < \gamma$, then $\widehat{E} = E^\star$ on the uniform-validation event. Indeed, for $e \in E^\star$,

$$\widehat{r}_e \geq r_e(P) - \epsilon_{n,\pi} > \tau, \tag{10}$$

whereas for $e \in E_\pi \setminus E^\star$,

$$\widehat{r}_e \le r_e(P) + \epsilon_{n,\pi} < \tau. \tag{11}$$

Consequently, $\Pr(\widehat{E} = E^\star) \ge 1 - \delta$ whenever

$$n > \frac{\log(2|E_\pi|/\delta)}{2\gamma^2}, \tag{12}$$

which proves Equation 5. Coverage is essential: if $E^\star \not\subseteq E_\pi$, no amount of validation data can recover an omitted relation.

Relation-error decomposition and its comparator. To isolate the effect of deleting irrelevant coordinates, define the full-coordinate plug-in estimator $\widetilde{r}_{\text{full}} = \widehat{r} = r(P) + \boldsymbol{\xi}$. This is a specific comparator that retains every empirical coordinate; it does not represent every generator that searches the full candidate space. On the recovery event, the prior-guided plug-in estimator is $\widetilde{r}_\pi = P_{E^\star}\widehat{r}$, where $P_{E^\star}$ is coordinate projection onto $E^\star$. Thus

$$\begin{aligned} e_{\text{full}}^2 &:= \|\widetilde{r}_{\text{full}} - r(P)\|_2^2 = \|\boldsymbol{\xi}\|_2^2, \\ e_\pi^2 &:= \|\widetilde{r}_\pi - r(P)\|_2^2 = \|P_{E^\star}\boldsymbol{\xi}\|_2^2, \end{aligned} \tag{13}$$

and orthogonality gives

$$e_{\text{full}}^2 - e_\pi^2 = \|P_{E_{\text{all}}\setminus E^\star}\boldsymbol{\xi}\|_2^2 \ge 0. \tag{14}$$

The inequality is strict only when finite observations create nonzero noise in at least one excluded coordinate. For a full-space procedure that also performs valid model selection, Proposition 1 establishes a smaller validation radius and sample requirement for E_π, but it does not by itself establish strict realized-error ordering.

Compilation error. Let Q_π and Q_{full} be distributions materialized from the two plug-in targets, and define $\eta_\pi = \|r(Q_\pi) - \widetilde{r}_\pi\|_2$ and $\eta_{\text{full}} = \|r(Q_{\text{full}}) - \widetilde{r}_{\text{full}}\|_2$. The triangle inequality yields

$$\Delta_{\text{rel}}(P, Q_\pi) \le e_\pi + \eta_\pi, \qquad \Delta_{\text{rel}}(P, Q_{\text{full}}) \le e_{\text{full}} + \eta_{\text{full}}. \tag{15}$$

Therefore relation selection gives a smaller certified upper bound when $e_\pi + \eta_\pi < e_{\text{full}} + \eta_{\text{full}}$. A sufficient condition for strict ordering of the actual relation shifts is the stronger inequality

$$e_\pi + \eta_\pi < e_{\text{full}} - \eta_{\text{full}}, \tag{16}$$

because the reverse triangle inequality gives $\Delta_{\text{rel}}(P, Q_{\text{full}}) \ge e_{\text{full}} - \eta_{\text{full}}$. These conditions formalize the qualification in the main paper that compilation error must not offset the relation-selection gain.

Theorem 1: downstream-risk bound. For a downstream model class $\mathcal{F}$, define $D_{\mathcal{F}}(P, Q) = \sup_{f\in\mathcal{F}} |R_P(f) - R_Q(f)|$. Assume

$$D_{\mathcal{F}}(P, Q) \le L_{\text{rel}}\Delta_{\text{rel}}(P, Q) + \varepsilon_{\text{high}}, \tag{17}$$

where $\varepsilon_{\text{high}}$ accounts for task-relevant structure not represented by r. Let $0 \le \ell \le B$, and let $\widehat{f}_{Q,N}$ be an empirical-risk minimizer trained on N samples that are i.i.d. from a fixed Q, conditional on the learned specification. Then, with probability at least $1 - \delta_{\text{gen}}$ over these samples,

$$\begin{aligned} R_P(\widehat{f}_{Q,N}) - \inf_{f\in\mathcal{F}} R_P(f) \le{}& 2L_{\text{rel}}\Delta_{\text{rel}}(P, Q) + 2\varepsilon_{\text{high}} \\ &+ 4\mathfrak{R}_N(\ell \circ \mathcal{F}) + 2B\sqrt{\frac{\log(2/\delta_{\text{gen}})}{2N}}. \end{aligned} \tag{18}$$

To prove the result, add and subtract risks under Q. The two domain-transfer terms are each at most $D_{\mathcal{F}}(P, Q)$, and standard Rademacher uniform convergence bounds the ERM excess risk under Q by the final two terms. Substitution of the assumed task-distance inequality gives the display and Equation 6. Together with relation recovery, both events hold with probability at least $1 - \delta - \delta_{\text{gen}}$ by a union bound.

The i.i.d. condition is an idealization. The fixed-quota implementation introduces dependent allocations and the deterministic $1/N$ proportion error in Equation 4; applying the displayed Rademacher bound directly to that implementation additionally requires a concentration result for its quota-conditioned sampler.

Corollary 2: target-size scaling. Let Q_N denote the distribution induced by executing a fixed specification at target size N. Assume target-size stability,

$$\Delta_{\mathrm{rel}}(P, Q_N) \le \overline{\Delta} + \frac{c}{N}, \tag{19}$$

where c/N includes deterministic allocation and rounding effects. Then the preceding theorem gives

$$\begin{aligned} R_P(\widehat{f}_{Q_N,N}) - \inf_{f\in\mathcal{F}} R_P(f) \le 2L_{\mathrm{rel}}\left(\overline{\Delta} + \frac{c}{N}\right) + 2\varepsilon_{\mathrm{high}} \\ + \mathrm{Est}_{\mathcal{F}}(N). \end{aligned} \tag{20}$$

When $\mathrm{Est}_{\mathcal{F}}(N) = O(N^{-1/2})$, increasing N decreases the finite-sample part and the rounding contribution, while the irreducible limit is controlled by $\overline{\Delta}$ and $\varepsilon_{\mathrm{high}}$. This result explains how a stable specification can support larger generated training sets; it does not assert that empirical test performance must increase monotonically for every dataset or optimizer.

Corollary 3: mixed-data expansion. If n real rows are augmented with m independent synthetic rows, let $\alpha = m/(n+m)$ and $M_{\alpha,Q} = (1-\alpha)P + \alpha Q$. Linearity of risk in the data distribution gives the exact identity

$$D_{\mathcal{F}}(P, M_{\alpha,Q}) = \alpha D_{\mathcal{F}}(P, Q). \tag{21}$$

Given a shift budget $\rho > 0$ and a valid upper bound $\overline{D}_Q \ge D_{\mathcal{F}}(P, Q)$, the certified synthetic proportion is

$$\alpha_{\max}(Q) = \min\left\{1, \frac{\rho}{\overline{D}_Q}\right\}, \tag{22}$$

with the convention $\alpha_{\max} = 1$ if $\overline{D}_Q = 0$. If $\overline{D}_\pi < \overline{D}_{\mathrm{full}}$ and $\rho < \overline{D}_{\mathrm{full}}$, then $\alpha_{\max}(Q_\pi) > \alpha_{\max}(Q_{\mathrm{full}})$. When $0 < \rho < \overline{D}_Q$, this is equivalently

$$m \le m_{\max}(Q) = \frac{n\rho}{\overline{D}_Q - \rho}. \tag{23}$$

This corollary concerns real–synthetic augmentation. The pure-synthetic target-size experiments in the main paper are governed by Corollary 2 instead.

Admissible compiled distributions. The validated relations and shared marginal estimates define

$$\begin{aligned} \mathcal{Q}(\widehat{E}) = \big\{Q : Q_j \in \mathcal{M}_{j,n},\ j = 1, \dots, p; \\ \|r_{\widehat{E}}(Q) - \widehat{r}_{\widehat{E}}\|_2 \le \eta_r; \\ Q(\mathcal{C}_{\mathrm{hard}}) = 1\big\}. \end{aligned} \tag{24}$$

Here $\mathcal{M}_{j,n}$ is the allowed marginal family, η_r is the relation-realization tolerance, and $\mathcal{C}_{\mathrm{hard}}$ is the hard feasible set. Unmeasured coordinates outside $\widehat{E}$ are not guaranteed to vanish unless the specification explicitly imposes the corresponding independence requirement. The executable specification is the finite representation used by PAVES to materialize a member of this family.

A.2 Constraint Families and Operator Semantics

The executable specification uses typed operators to connect empirical constraint extraction with dataset generation. The four families introduced in the main paper determine the scope of each constraint, while the operator subtype determines what evidence is extracted, how the constraint is validated, and how it is executed after compilation. Domain operators restrict feasible values; statistical operators parameterize univariate sampling targets; relational operators represent conditional associations and co-occurrence among columns in one table; and derived operators materialize validated field dependencies. Every executable operator also defines a target-size scaling rule and a fallback for unsupported or infeasible cases. Table 4 summarizes the detailed operator branches used by PAVES.

Table 4: Detailed constraint operators, extracted evidence, and compiled behaviors.

Family	Operator branch	Extracted or validated evidence	Compiled behavior
Domain	Domain	Observed categorical support	Discrete feasible set
Domain	Range	Validated numerical support and bounds	Bounded feasibility predicate
Domain	Missingness	Missing rate and conditional missingness	Missing-value allocation
Statistical	Marginal	Frequencies, quantiles, or moments	Smoothed univariate sampling
Relational	ConditionalDistribution	Support, effect, and held-out gain	Conditional sampling with sparse backoff
Relational	Cooccurrence	Joint support and association strength	Joint or conditional value sampling
Derived	FunctionalDependency	Support and dependency violation rate	Validated deterministic mapping
Derived	DerivedField	Residual error under a tolerance	Registered field transformation

A.3 Constraint Extraction and Consolidation Details

Operator-specific validation uses dependency violation rate, derived-field residual error, categorical association strength, or held-out conditional likelihood gain. Tests in the same candidate family are corrected for multiple comparisons; operators without a meaningful significance test use a held-out error threshold. Conditional profiles choose the split with the largest penalized held-out gain, retain at most the top-k frequent values, merge remaining values into OTHER, and require support m in each child. Splitting stops at non-positive gain or depth d, and sparse leaves back off to the nearest validated parent.

The validated pool is consolidated by the objective

$$\mathcal{C}^* = \arg\max_{\mathcal{C}' \subseteq \mathcal{C}_{\text{val}}} \Big[\lambda_{\text{fit}} \operatorname{Fit}(\mathcal{C}') - \lambda_{\text{comp}} \operatorname{Complexity}(\mathcal{C}') - \lambda_{\text{red}} \operatorname{Redundancy}(\mathcal{C}') - \lambda_{\text{conf}} \operatorname{Conflict}(\mathcal{C}') \Big]. \tag{25}$$

Here, fit measures agreement with the extracted relations, complexity penalizes constraint count and conditioning depth, redundancy penalizes equivalent coverage, and conflict measures incompatible domains, allocations, or deterministic requirements. Greedy consolidation inserts structural requirements first and then orders remaining candidates by incremental objective gain; semantic confidence only breaks ties. Static checks detect type and interval contradictions, integer feasibility detects allocation conflicts, and the feasibility backend detects remaining hard conflicts. An incompatible candidate is rejected or, when its operator permits approximation, converted to a soft statistical target.

A.4 Constraint-Guided Generation Procedure

Algorithm 1 expands the compilation and execution stages described in the main paper. The procedure first constructs semantic guidance and global constraints, partitions the source rows by the designated target columns, and estimates local conditional constraints within each stratum. The extraction calls include the empirical validation and consistency-aware consolidation defined in Section 3.2. When no target column is designated, the stratification contains one group ($K = 1$). For stratum k, $\mathbf{Z}_k \in \mathbb{R}^{N_k \times p}$ denotes the latent-sample matrix whose rows $\mathbf{z}_{ki}$ are independently drawn from $\mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. These rank-correlated latent samples determine marginal quota allocations, while the feasibility backend checks the directly encodable constraints in batches. The algorithm shows the Z3 backend used in our implementation; the executable specification itself remains independent of this backend.

B Dataset

B.1 Standard classification and regression datasets

The standard tabular benchmark contains eleven real-world datasets spanning industrial inspection, remote sensing, astronomy, biology, health, commerce, transportation, and online media. Eight datasets define classification tasks. Faults identifies steel-plate fault categories, Wilt distinguishes diseased from healthy trees, HTRU2 identifies pulsar candidates, Bean predicts dry-bean varieties,

Algorithm 1 Constraint-Guided Tabular Data Generation with Z3

Require: Source table $\mathcal{D}$, schema $\mathcal{S}$, optional target columns $\mathcal{Y}$, task type τ, target size N
Ensure: Synthetic table $\widetilde{\mathcal{D}}$ with N rows

$\mathcal{H} \leftarrow \textsc{SemanticProfile}(\mathcal{S}, \mathcal{D}_{1:30}, \mathcal{Y}, \tau)$
$\mathcal{C}_{\text{global}} \leftarrow \textsc{ExtractMarginalConstraints}(\mathcal{D}, \mathcal{H})$
$\{\mathcal{D}_k\}_{k=1}^{K} \leftarrow \textsc{TargetStratification}(\mathcal{D}, \mathcal{Y}, \tau)$
$\{N_k\}_{k=1}^{K} \leftarrow \textsc{AllocateStratumSizes}(\{|\mathcal{D}_k|\}_{k=1}^{K}, N)$
for $k = 1, \ldots, K$ **do**
    $\mathcal{C}_k \leftarrow \textsc{ExtractConditionalConstraints}(\mathcal{D}_k, \mathcal{H})$
    $\mathbf{R}_k \leftarrow \textsc{EstimateRankCorrelation}(\mathcal{D}_k)$
    $\mathbf{Z}_k \leftarrow \textsc{SampleGaussianLatents}(N_k, \mathbf{R}_k)$
    $\mathcal{A}_k \leftarrow \textsc{AllocateMarginalQuotas}(\mathbf{Z}_k, \mathcal{C}_k, N_k)$
    for each batch $\mathcal{B} \subseteq \mathcal{A}_k$ **do**
        $\Phi_{\mathcal{B}} \leftarrow \textsc{EncodeZ3}(\mathcal{B}, \mathcal{C}_{\text{global}}, \mathcal{C}_k)$
        $\textsc{Assert}(\Phi_{\mathcal{B}},$ domain, type, missingness, marginal, and joint constraints$)$
        $\sigma_{\mathcal{B}} \leftarrow \textsc{Z3Check}(\Phi_{\mathcal{B}})$
        if $\sigma_{\mathcal{B}} = \text{SAT}$ **then**
            $\mathbf{m}_{\mathcal{B}} \leftarrow \textsc{Z3Model}$
            $\widetilde{\mathcal{B}} \leftarrow \textsc{Decode}(\mathbf{m}_{\mathcal{B}})$
        else
            return generation failure
        end if
    end for
    $\widetilde{\mathcal{D}}_k \leftarrow \textsc{Concatenate}(\{\widetilde{\mathcal{B}}\})$
end for
$\widetilde{\mathcal{D}}' \leftarrow \textsc{Concatenate}(\{\widetilde{\mathcal{D}}_k\}_{k=1}^{K})$
$\widetilde{\mathcal{D}} \leftarrow \textsc{Shuffle}(\widetilde{\mathcal{D}}')$
return $\widetilde{\mathcal{D}}$

Obesity predicts obesity levels, Titanic predicts passenger survival, Shoppers predicts online purchase intention, and Indian Liver Patient identifies liver-disease cases. The remaining three datasets define regression tasks: News predicts online article popularity, Insurance predicts individual insurance charges, and Abalone predicts age from physical measurements. Dataset sizes range from 579 to 39,644 records, while the number of numerical columns ranges from 4 to 60. This variation provides a heterogeneous benchmark for evaluating whether a single synthesis framework can retain useful statistical structure across different sample sizes, feature dimensions, and prediction objectives. Table 5 summarizes the dataset statistics.

Table 5: Summary of the eleven real-world datasets used for standard tabular synthesis experiments.

Dataset	Rows	Task	Numerical columns
Faults	1,941	Classification	34
Wilt	4,839	Classification	5
HTRU2	17,898	Classification	8
News	39,644	Regression	60
Bean	13,611	Classification	16
Obesity	2,111	Classification	8
Titanic	714	Classification	6
Insurance	1,338	Regression	4
Abalone	4,177	Regression	8
Shoppers	12,330	Classification	16
Indian Liver Patient	579	Classification	9

Table 6: Sources and download links for the standard tabular benchmark datasets.

Dataset	Source	Task	Download link
Abalone	OpenML	Regression	https://www.openml.org/search?type=data&sort=runs&id=183&status=active
Bean	UCI	Multiclass classification	https://archive.ics.uci.edu/dataset/602/dry+bean+dataset
Faults	UCI	Multiclass classification	https://archive.ics.uci.edu/dataset/198/steel+plates+faults
HTRU2	UCI	Binary classification	https://archive.ics.uci.edu/dataset/372/htru2
Indian Liver Patient	Kaggle	Binary classification	https://www.kaggle.com/datasets/uciml/indian-liver-patient-records?resource=download
Insurance	Kaggle	Regression	https://www.kaggle.com/datasets/mirichoi0218/insurance
News	UCI	Regression	https://archive.ics.uci.edu/dataset/332/online+news+popularity
Obesity	Kaggle	Multiclass classification	https://www.kaggle.com/datasets/tathagatbanerjee/obesity-dataset-uci-ml
Shoppers	Kaggle	Binary classification	https://www.kaggle.com/datasets/henrysue/online-shoppers-intention
Titanic	Kaggle	Binary classification	https://www.kaggle.com/c/titanic/data
Wilt	OpenML	Binary classification	https://www.openml.org/search?type=data&sort=runs&id=40983&status=active

B.2 Dataset Download Links

Table 6 lists the repositories from which the eleven standard tabular datasets can be obtained. The task type follows the experimental setting used in this paper.

C Experiments

C.1 Per-Dataset Statistical Fidelity

Table 7 reports the statistical and specification fidelity of PAVES on each of the eleven standard tabular benchmarks. The mean values correspond to the PAVES column in Table 1 of the main paper.

C.2 Per-Dataset Downstream Classification

Table 8 decomposes the downstream classification results of PAVES reported in Table 2 of the main paper. Each value is the TSTR accuracy obtained by training the corresponding classifier on data generated by PAVES and evaluating it on the untouched real test split.

C.3 Per-Run Results for Scalable Dataset Expansion

The scaling experiment contains eight classification datasets, four target-size multipliers, and three downstream Transformer training seeds, for 96 completed runs. Seeds 42, 43, and 44 control downstream optimization; the synthetic table is fixed across these seeds for each dataset–multiplier pair.

Table 7: Per-dataset statistical and specification fidelity of PAVES on the eleven standard tabular benchmarks.

Dataset	Precision ↑	Recall ↑	Hard Constraint Satisfaction ↑	Conditional Distribution Fidelity ↑	Statistical Target Error ↓
Abalone	0.991199	0.377304	1.000000	0.987722	0.000415
Bean	0.994830	0.415776	1.000000	0.995497	0.000410
Faults	0.964648	0.151228	1.000000	0.985485	0.000616
HTRU2	0.978036	0.358301	1.000000	0.996497	0.001274
Indian Liver	0.982784	0.380887	1.000000	0.991521	0.001212
Insurance	0.980759	0.471450	1.000000	0.986928	0.000861
News	0.865420	0.046534	1.000000	0.991527	0.001574
Obesity	0.966469	0.128723	1.000000	0.992274	0.000576
Shoppers	0.973577	0.312036	1.000000	0.993203	0.002572
Titanic	0.951032	0.319514	1.000000	0.980499	0.002153
Wilt	0.987617	0.402177	1.000000	0.992861	0.003455
Mean	**0.966943**	**0.305812**	**1.000000**	**0.990365**	**0.001374**

Table 8: Per-dataset TSTR classification accuracy of PAVES on the eight standard classification benchmarks.

Dataset	Decision Tree	Random Forest	XGBoost	CatBoost
Bean	0.913599	0.923591	0.926897	0.929616
Faults	0.589387	0.676455	0.679547	0.676971
HTRU2	0.977595	0.976813	0.976813	0.977037
Indian Liver	0.740995	0.759863	0.747856	0.763293
Obesity	0.892942	0.898626	0.917101	0.907627
Shoppers	0.893998	0.891646	0.888483	0.894404
Titanic	0.809203	0.789001	0.786756	0.799102
Wilt	0.982021	0.981401	0.983261	0.983468
Mean	**0.849968**	**0.862175**	**0.863339**	**0.866440**
Sample SD	0.132557	0.109309	0.111995	0.109198

Checkpoint selection uses only the synthetic validation split. Preprocessing is fitted on the synthetic training split, and the fixed real evaluation set is accessed once after the selected checkpoint is reloaded. Tables 9–12 report every run. The main paper uses Macro-F1 for Figure 4; the remaining metrics are included here to expose class-imbalance effects and dataset-level heterogeneity. These three seeds quantify downstream training variability rather than variability in specification extraction or synthetic-table generation. The experiment uses a fixed Transformer configuration and training protocol across datasets, and its real evaluation table is a fixed benchmark set rather than an independently resampled population. We therefore interpret the results as measured downstream scaling under this protocol, without attributing the gains to information beyond that contained in the source data.

Table 9: Per-dataset and per-seed Transformer results for synthetic training data generated at $1\times$ the source size.

Dataset	Seed	Accuracy	Macro-F1	Weighted F1	Balanced Accuracy
Bean	42	0.9337	0.9445	0.9338	0.9415
Bean	43	0.9334	0.9439	0.9334	0.9439
Bean	44	0.9390	0.9496	0.9392	0.9476
Faults	42	0.8547	0.8632	0.8542	0.8663
Faults	43	0.7723	0.7848	0.7721	0.8029
Faults	44	0.8733	0.8873	0.8734	0.8951
HTRU2	42	0.9806	0.9408	0.9805	0.9336
HTRU2	43	0.9804	0.9392	0.9801	0.9261
HTRU2	44	0.9800	0.9372	0.9795	0.9182
Indian Liver	42	0.7152	0.4170	0.5956	0.5003
Indian Liver	43	0.7152	0.4172	0.5967	0.5010
Indian Liver	44	0.7146	0.4165	0.5963	0.4987
Obesity	42	0.9858	0.9853	0.9858	0.9853
Obesity	43	0.9773	0.9767	0.9773	0.9765
Obesity	44	0.9829	0.9823	0.9830	0.9821
Shoppers	42	0.9062	0.8113	0.9037	0.7961
Shoppers	43	0.9053	0.8117	0.9034	0.7999
Shoppers	44	0.9078	0.8226	0.9075	0.8204
Titanic	42	0.8109	0.7943	0.8053	0.7852
Titanic	43	0.8039	0.7803	0.7938	0.7695
Titanic	44	0.8123	0.7990	0.8087	0.7919
Wilt	42	0.9895	0.9471	0.9893	0.9366
Wilt	43	0.9859	0.9272	0.9855	0.9041
Wilt	44	0.9874	0.9388	0.9875	0.9428

Table 10: Per-dataset and per-seed Transformer results for synthetic training data generated at $2\times$ the source size.

Dataset	Seed	Accuracy	Macro-F1	Weighted F1	Balanced Accuracy
Bean	42	0.9409	0.9514	0.9408	0.9508
Bean	43	0.9495	0.9591	0.9493	0.9586
Bean	44	0.9569	0.9661	0.9570	0.9649
Faults	42	0.9423	0.9500	0.9421	0.9561
Faults	43	0.9382	0.9483	0.9382	0.9513
Faults	44	0.8861	0.9055	0.8845	0.9037
HTRU2	42	0.9808	0.9408	0.9806	0.9288
HTRU2	43	0.9808	0.9402	0.9804	0.9252
HTRU2	44	0.9809	0.9407	0.9806	0.9269
Indian Liver	42	0.6857	0.6343	0.6932	0.6435
Indian Liver	43	0.6390	0.6305	0.6546	0.7038
Indian Liver	44	0.7029	0.6257	0.6988	0.6228
Obesity	42	0.9938	0.9937	0.9938	0.9937
Obesity	43	0.9910	0.9907	0.9910	0.9906
Obesity	44	0.9957	0.9956	0.9957	0.9956
Shoppers	42	0.9535	0.9102	0.9533	0.9055
Shoppers	43	0.9099	0.8220	0.9084	0.8116
Shoppers	44	0.9769	0.9558	0.9769	0.9551
Titanic	42	0.8389	0.8294	0.8370	0.8235
Titanic	43	0.8095	0.7930	0.8039	0.7840
Titanic	44	0.8319	0.8240	0.8310	0.8209
Wilt	42	0.9895	0.9490	0.9895	0.9547
Wilt	43	0.9903	0.9513	0.9902	0.9407
Wilt	44	0.9903	0.9537	0.9904	0.9660

Table 11: Per-dataset and per-seed Transformer results for synthetic training data generated at $5\times$ the source size.

Dataset	Seed	Accuracy	Macro-F1	Weighted F1	Balanced Accuracy
Bean	42	0.9928	0.9948	0.9928	0.9953
Bean	43	0.9910	0.9929	0.9910	0.9930
Bean	44	0.9921	0.9937	0.9921	0.9941
Faults	42	0.9974	0.9984	0.9974	0.9986
Faults	43	0.9938	0.9952	0.9938	0.9959
Faults	44	0.9923	0.9932	0.9923	0.9946
HTRU2	42	0.9821	0.9453	0.9819	0.9386
HTRU2	43	0.9815	0.9439	0.9814	0.9410
HTRU2	44	0.9818	0.9453	0.9818	0.9442
Indian Liver	42	0.6960	0.6284	0.6966	0.6289
Indian Liver	43	0.7098	0.6692	0.7191	0.6859
Indian Liver	44	0.6943	0.6321	0.6972	0.6350
Obesity	42	0.9986	0.9985	0.9986	0.9985
Obesity	43	0.9995	0.9995	0.9995	0.9995
Obesity	44	0.9991	0.9990	0.9991	0.9990
Shoppers	42	0.9970	0.9943	0.9970	0.9950
Shoppers	43	0.9974	0.9951	0.9974	0.9965
Shoppers	44	0.9984	0.9969	0.9984	0.9963
Titanic	42	0.8613	0.8559	0.8611	0.8549
Titanic	43	0.8249	0.8145	0.8228	0.8090
Titanic	44	0.8683	0.8628	0.8680	0.8608
Wilt	42	0.9913	0.9575	0.9913	0.9575
Wilt	43	0.9944	0.9728	0.9944	0.9754
Wilt	44	0.9919	0.9604	0.9919	0.9596

Table 12: Per-dataset and per-seed Transformer results for synthetic training data generated at $10\times$ the source size.

Dataset	Seed	Accuracy	Macro-F1	Weighted F1	Balanced Accuracy
Bean	42	0.9985	0.9988	0.9985	0.9988
Bean	43	0.9965	0.9973	0.9965	0.9975
Bean	44	0.9992	0.9994	0.9992	0.9994
Faults	42	0.9992	0.9994	0.9993	0.9997
Faults	43	0.9988	0.9994	0.9987	0.9995
Faults	44	0.9985	0.9991	0.9985	0.9994
HTRU2	42	0.9825	0.9462	0.9823	0.9358
HTRU2	43	0.9821	0.9444	0.9818	0.9298
HTRU2	44	0.9817	0.9440	0.9816	0.9356
Indian Liver	42	0.9988	0.9968	0.9981	0.9970
Indian Liver	43	0.9984	0.9988	0.9986	0.9988
Indian Liver	44	0.9977	0.9981	0.9982	0.9970
Obesity	42	1.0000	1.0000	1.0000	1.0000
Obesity	43	1.0000	1.0000	1.0000	1.0000
Obesity	44	1.0000	1.0000	1.0000	1.0000
Shoppers	42	0.9983	0.9968	0.9983	0.9979
Shoppers	43	0.9992	0.9984	0.9992	0.9976
Shoppers	44	0.9988	0.9977	0.9988	0.9976
Titanic	42	0.9104	0.9043	0.9088	0.8946
Titanic	43	0.9048	0.8996	0.9039	0.8937
Titanic	44	0.8922	0.8835	0.8894	0.8716
Wilt	42	0.9932	0.9672	0.9932	0.9765
Wilt	43	0.9936	0.9684	0.9936	0.9659
Wilt	44	0.9948	0.9745	0.9948	0.9702