ZERO- AND FEW-SHOT ARBITRARY CONDITIONING WITH SEMANTIC AND PREDICTIVE CONTEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalist statistical inference requires answering conditional queries whose targets and observed features may change across previously unseen schemas. We study a zero- and few-shot arbitrary conditioning problem over semantically described tabular datasets, where observed feature–value pairs alone may leave the target distribution ambiguous. We introduce the **Arbitrary Set-based Permutation-Invariant Reasoning Engine (ASPIRE)**, which draws on feature and dataset semantics, optional support examples, and external predictive context. ASPIRE represents observed values and external forecast probabilities as semantically grounded atoms, contextualizes co-occurring atoms with a shared permutation-equivariant module, and combines distinct sources of evidence through role-aware late fusion. Target-native likelihood heads return categorical probability mass functions or continuous densities; the same conditional interface supports chain-rule table generation. Trained on over 1,400 real-world datasets, ASPIRE leads the evaluated zero-shot comparisons. With five support examples, it achieves 12.7% higher F1 and 5.9% lower normalized RMSE than the strongest competing methods. External predictive context yields further gains, while the standalone conditional model also achieves the strongest five-shot table-generation fidelity among the evaluated methods.

1 INTRODUCTION

The rise of generalist AI models has expanded the range of tasks that a single model can address (Brown et al., 2020; Sanh et al., 2022; Chung et al., 2024). Here, we study a generalist *statistical inference* problem spanning a broad universe of features and conditional tasks. At inference time, the model is given a target feature and a collection of observed feature values for an instance, and must infer the target's conditional distribution. Both the target and the observations may be drawn from the same broad feature universe. This calls for a single model that can reason about statistical relationships among features and their values across diverse instances (Fig. 1).

Arbitrary conditioning methods (Douglas et al., 2017; Ivanov et al., 2018; Li et al., 2020; Strauss & Oliva, 2021) train a single model to answer many conditional queries: given any subset of observed features, infer the distribution of a chosen target. These methods have largely operated in a *closed-world* setting, where observations and targets are restricted to the d features of a fixed dataset schema known during training. They therefore generalize across observed–target partitions of those d dimensions. We instead seek to infer across previously unseen schemas whose features are specified through semantic descriptions.

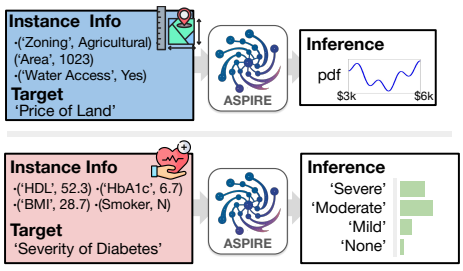

Figure 1: Illustration of zero-shot arbitrary conditioning, where a single ASPIRE model answers support-free conditional queries for different targets under different observed feature subsets, including variables not directly encountered during training.

Cross-tabular transfer methods (Hollmann et al., 2022; QU et al., 2025; Zhang et al., 2025) typically rely on many labeled rows from the target

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.

dataset, either for fine-tuning or as in-context examples. We instead enable *direct conditional inference* on an unseen schema with no labeled rows, while allowing support examples when available.

Without labeled rows from the target dataset, a set of observed feature–value pairs may leave the underlying instance and its target distribution highly ambiguous. Arbitrary conditioning across unseen schemas therefore requires more than cross-table transfer or masked-feature prediction: it requires conditional reasoning over semantically grounded evidence. We develop a contextual framework that conditions on feature and target semantics, dataset context, a handful of support examples, and external predictions when available to help resolve this ambiguity.

To reason over these distinct sources of context, we develop the **Arbitrary Set-based Permutation-Invariant Reasoning Engine (ASPIRE)**. ASPIRE represents feature–value pairs as semantically grounded *atoms* and contextualizes co-occurring atoms with a shared permutation-equivariant module. The module is reused across query and support instances and, when predictive context is available, *belief atoms* that pair candidate outcomes with probabilities from an external model. A role-aware late-fusion stage combines query evidence, support evidence, external belief, and textual dataset and target context into a permutation-invariant representation, preserving the structure of each source of evidence before aggregation. A target-native likelihood interface returns probability mass functions for categorical targets and probability density functions for continuous targets. When an external belief is provided, ASPIRE also predicts a per-query mixing coefficient that determines its contribution to the final distribution. This zero- and few-shot arbitrary conditioning interface supports classification, regression, and chain-rule inference over unobserved variables, given observed feature–value sets from unseen schemas.

Our contributions are: (1) We formulate zero- and few-shot arbitrary conditioning across unseen schemas, with the target and observed variables specified at inference time and semantic descriptions, support examples, and external predictions available as context. **(2)** We introduce a common atom-set representation for heterogeneous evidence: observed values and predictive contexts remain paired with the features or outcomes they describe. A shared permutation-equivariant set-to-set module captures relationships among co-occurring atoms while preserving their individual identities, and is reused across query instances, support examples, and external predictive beliefs. **(3)** We develop a role-aware late-fusion architecture that combines these separately contextualized sources with dataset and target semantics to answer each query. Its target-native likelihood interface produces categorical probabilities or continuous densities from the same model. **(4)** We incorporate an external forecast both as conditioning information and through a learned query-specific mixture coefficient; ablations distinguish the benefit of conditioning from output mixing. **(5)** We benchmark ASPIRE against a broad suite of tabular methods on held-out schemas, where it achieves the strongest zero- and five-shot classification and regression performance. External predictive context yields further gains, and five-shot table generation achieves the strongest distributional fidelity.

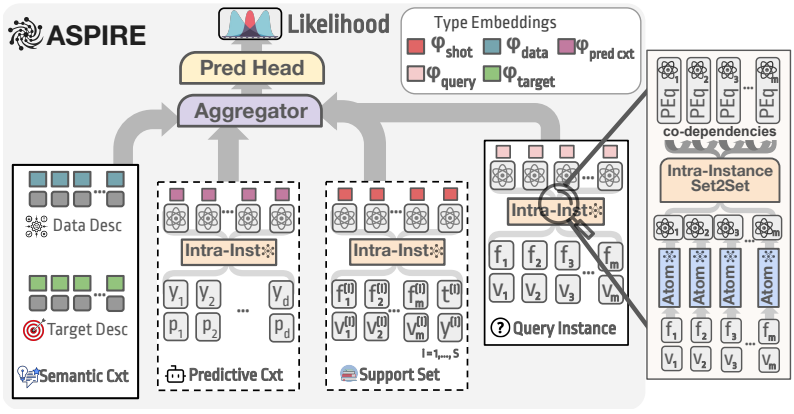

Figure 2: ASPIRE model processes query instances, optional support sets and predictive context, and dataset context through set-based mappings, maintaining permutation invariance at both feature and instance levels.

2 METHOD

We formalize zero- and few-shot arbitrary conditioning in a generalist statistical inference setting for conditional queries that must be answered across heterogeneous data, with target and observed variables identified through semantic descriptions and only partially encountered during training.

A successful model should: (i) represent partially observed instances while preserving the unordered structure of co-occurring feature–value evidence, (ii) combine query evidence, optional support examples, semantic context, and external predictive context without collapsing these sources too early, and (iii) return target-native conditional likelihoods rather than only point predictions. These requirements motivate the core design of ASPIRE Fig. 2: a shared permutation-equivariant intra-instance set contextualizer, a role-aware late-fusion aggregator, and a multi-domain conditional likelihood interface.

Together, these components, parameterized by γ, define a single conditional estimator

$$p_\gamma(v_t \mid x_o, f_t, S, c, p_{\mathrm{pc}}),$$

where f_t denotes the target feature metadata, including its textual description, type, and domain; v_t is the corresponding target value; $x_o = \{(f_m, v_m)\}_{m \in o}$ is the unordered set of observed feature–value pairs; S is an optional support set from the same dataset; c denotes semantic context describing the dataset and target; and p_{pc} denotes optional predictive context supplied by an external model.

2.1 SHARED INTRA-INSTANCE SET CONTEXTUALIZER

We represent each partially observed example as an unordered set of feature-value atoms,

$$e = \{(f_m, v_m)\}_{m=1}^{M},$$

where each feature f_m contains semantic metadata such as textual description and type information. ASPIRE first maps each atom to a learned semantically grounded embedding

$$\psi(f_m, v_m),$$

which interprets the value v_m relative to the semantics of feature f_m rather than only its raw magnitude or dataset-local position; Appendix A.4 gives the detailed construction of ψ, which utilizes an MLP atop of domain-specific embeddings of values (Fourier-based for real values, language-embeddings for discrete values), along with language-embeddings of their respective descriptions.

Intra-Instance Set-to-Set Module. One of our central contributions is to process atom embeddings with a *shared permutation-equivariant set-to-set contextualization module* over the co-occurring atoms of an instance, $\rho(e)$. Rather than serializing the instance or immediately compressing it to a single vector, this module maps the input atom set to a contextualized output set of the same cardinality,

$$\{\psi(f_m, v_m)\}_{m=1}^{M} \;\mapsto\; \{\rho_m(e)\}_{m=1}^{M}, \tag{1}$$

where each output embedding $\rho_m(e)$ remains associated (i.e., corresponds) with a particular atom while depending (and being informed by) on the other co-occurring atoms in the same instance. This yields a structured set representation of the instance that preserves per-atom structure, respects feature-order equivariance, and avoids an early compression bottleneck.

When few-shot support is available, each support example

$$e^{(s)} = \{(f_m, v_m^{(s)})\}_{m=1}^{M_s} \cup \{(f_t, v_t^{(s)})\}$$

is processed using the same atom and intra-instance modules as the query instance. This yields

$$\Lambda(S) = \{\rho(e^{(s)})\}_{s=1}^{|S|},$$

with zero-shot inference when $S = \varnothing$. Query and support examples are therefore mapped into a common atom-level representation space before any cross-source aggregation occurs.

Predictive Context. An external model may already hold a belief about the target, expressed as a distribution $p_{\mathrm{pc}}(v_t)$. We call this *predictive context* and propose to reason over it as an additional

source of information. In practice, it may come from another cross-tabular method (§3). A central challenge is to represent this belief in a way that generalizes across arbitrary targets while respecting symmetries, such as the unordered nature of categorical target values. A fixed probability vector presumes a canonical ordering of values and loses the association between each probability and the value it describes.

We address this challenge using the same *atom* representation and *intra-instance set-to-set* module. Specifically, we express predictive context as a set of *belief atoms*, one per possible target value,

$$e^{\mathrm{pc}} = \{(f_t, y_k, p_k)\}_{k=1}^{K}, \qquad p_k = p_{\mathrm{pc}}(v_t = y_k \mid x_o, S),$$

and process them with the shared intra-instance set-to-set module (Eq. 1) as $\{\psi_{\mathrm{pred}}(\psi(f_t, y_k),\, p_k)\}_{k=1}^{K} \mapsto \{\rho_m(e^{\mathrm{pc}})\}_{k=1}^{K}$, where ψ_{pred} is an MLP and ψ is the atom embedding used for feature values. For a continuous target, we instead use quantiles at levels $\tau_1, \ldots, \tau_J$, each forming a belief atom, $e^{\mathrm{pc}} = \{(f_t, \tau_j, q_{\tau_j})\}_{j=1}^{J}$, and process the resulting set with the same intra-instance module.

2.2 Role-Aware Late-Fusion Aggregator

After contextualizing atoms within each instance, ASPIRE performs a second stage of set reasoning across information sources. A query consists of a partially observed instance

$$e_q = \{(f_m, v_m)\}_{m \in o} \cup \{(f_t, \varnothing)\},$$

a target feature description f_t, dataset and target text context $\mathbf{C}_{\text{data}}$ and $\mathbf{C}_{\text{targ}}$, and optionally a small support set and optionally predictive context. ASPIRE aggregates five typed elements in a shared inference space: observed query evidence, the target slot to be predicted, support-example evidence, semantic context and predictive context.

To enable this joint reasoning, each element is tagged with a *learnable role embedding* indicating whether it corresponds to query evidence, the target slot, support examples, predictive or semantic context. We additionally use a dedicated `CLS` summary token to collect the final global representation:

$$\begin{aligned}
\mathcal{T}_{\text{query}} &= \{[\rho_m(e_q), \varphi_{\text{query}}]\}_{m \in o} \cup \{[\rho_t(e_q), \texttt{CLS}]\}, \\
\mathcal{T}_{\text{support}} &= \textstyle\bigcup_{s=1}^{|S|} \{[\rho_m(e^{(s)}), \varphi_{\text{shot}}]\}_{m=1}^{M_s}, \\
\mathcal{T}_{\text{context}} &= \{[c_t, \varphi_{\text{data}}] : c_t \in \mathbf{C}_{\text{data}}\} \cup \{[c_t, \varphi_{\text{target}}] : c_t \in \mathbf{C}_{\text{targ}}\}. \\
\mathcal{T}_{\text{pred}} &= \{[\rho_k(e^{\mathrm{pc}}), \varphi_{\text{pred}}]\}_{k=1}^{|e^{\mathrm{pc}}|}
\end{aligned} \tag{2}$$

Note that we take text tokens c_t to already include text based positional embeddings, thus preserving semantic meaning within $\mathbf{C}_{\text{data}}$ and $\mathbf{C}_{\text{targ}}$, respectively. The role embeddings, φ, allow ASPIRE to reason jointly over evidence, task specification, support, and schema context without collapsing them into an undifferentiated sequence.

The resulting elements are aggregated through stacked Set Transformer layers (Lee et al., 2019):

$$r_t = \text{InterAggr}(\mathcal{T}_{\text{query}} \cup \mathcal{T}_{\text{support}} \cup \mathcal{T}_{\text{context}} \cup \mathcal{T}_{\text{pred}}), \tag{3}$$

where the final output corresponding to the `CLS` token serves as a global target-conditioned summary of all currently available information.

This late-fusion design preserves a distinction between two levels of reasoning. The intra-instance contextualizer models dependencies among co-occurring atoms within each example, while the aggregation module performs role-aware inference across query evidence, support examples, and semantic context. The resulting invariance claim applies to the *tabular set structure*: reordering observed feature-value atoms, support examples or predictive context does not change the computation. By contrast, dataset and target descriptions are encoded as ordinary text sequences, since token order is part of their semantic content.

2.3 Multi-Domain Conditional Likelihood Interface

ASPIRE is designed to answer arbitrary conditional queries whose targets may belong to different domains. Accordingly, it produces *target-native conditional likelihoods* rather than task-specific point

predictions. Given observed evidence x_o, target feature f_t, optional support set S, and semantic context c, ASPIRE estimates $p_\gamma(v_t \mid x_o, f_t, S, c)$. For categorical targets, ASPIRE returns a probability mass function over admissible target values. For continuous targets, it returns a probability density function in the native target domain. This distinction is central to the inference interface. The goal is not merely to support both classification and regression under a shared backbone, but to expose a unified conditional prediction mechanism that supports arbitrary targets together.

For a categorical target, ASPIRE computes logits via dot products between r_t (eq. 3) and language embeddings of the possible category values. For a categorical feature f_t with possible values $\{y_1, \ldots, y_K\}$,

$$p(v_t = y_k \mid r_t) = \frac{\exp(r_t^\top \mathrm{LE}(y_k))}{\sum_{k'=1}^{K} \exp(r_t^\top \mathrm{LE}(y_{k'}))},$$

where $\mathrm{LE}(y_k)$ a language embedding.

For a continuous target, ASPIRE predicts a mixture density rather than a scalar point estimate:

$$p(v_t \mid r_t) = \sum_{i=1}^{I} \omega_i \mathcal{N}(v_t; \mu_i, \sigma_i^2),$$

where $\langle \omega_i, \mu_i, \sigma_i \rangle_{i=1}^{I} = \mathrm{ContHead}_\theta(r_t)$ are produced by learned MLP layers. Using target-type-specific native-domain heads in this way avoids forcing continuous targets into tokenized surrogates and gives the same backbone a unified conditional interface across discrete and continuous tasks.

Dynamic External Prediction Integration. Besides serving as input context, an available external prediction can contribute directly to the final conditional likelihood. ASPIRE outputs a query-specific mixing coefficient $\alpha(r_t) \in [0, 1]$ and forms

$$p_\star(v_t \mid x_o, f_t, S, c, p_{\mathrm{pc}}) = \alpha_t \, p_\gamma(v_t \mid x_o, f_t, S, c, p_{\mathrm{pc}}) + (1 - \alpha_t) \, p_{\mathrm{pc}}(v_t \mid x_o, S), \tag{4}$$

where $\alpha_t = \alpha(r_t)$. Because the mixture defines the final prediction, p_γ need not act as a standalone predictor. It can use the external forecast as context and contribute where useful—whether by reinforcing its prediction or shifting probability toward alternatives—while the query-specific α_t determines the contribution of each distribution. When no predictive context is available, we set $\alpha_t = 1$.

Training. ASPIRE is trained to answer arbitrary conditional queries across heterogeneous datasets, with varying target features, observation patterns, support-set sizes, and availability of predictive context. Let $\mathcal{D} = \{D_d\}_{d=1}^{D}$ denote the training datasets, each $D_d = \{e_n\}_{n=1}^{N_d}$ consisting of instances $e_n = \{(f_m, v_m)\}_{m=1}^{M_n}$ drawn from its own schema, with every feature carrying a description, a type, and, when applicable, admissible categorical values. These descriptions are what allow one conditional prediction interface to be applied across schemas.

A training episode $\tau \sim \mathcal{T}$ selects a dataset, a query instance, a target feature f_t, an observed subset o excluding the target, and an optional support set S drawn from the same dataset excluding the query. With probability η_{pc} the episode also carries predictive context $p_{\mathrm{pc}}(v_t \mid x_o, f_t, S, c)$ otherwise $p_{\mathrm{pc}} = \varnothing$. The query target value v_t is withheld from the input and used for supervision.

The parameters γ, which include the mixing head, are optimized by the expected negative log-likelihood of the returned mixture,

$$\mathcal{L}(\gamma) = \mathbb{E}_{\tau \sim \mathcal{T}} \Big[-\log \big(p_\star(v_t \mid x_o, f_t, S, c, p_{\mathrm{pc}}) \big) \Big], \tag{5}$$

with the external model held fixed, and $p_\star$ as in eq. 4. Varying the target and observed subset exposes the model to different conditional queries, varying S trains it to use examples when available, and withholding p_{pc} on a fraction of episodes keeps p_γ a complete predictor on its own.

Joint Generation. The same conditional likelihood interface also supports generation over unobserved variables via the chain rule. Let $f_1, \ldots, f_d$ denote an ordering of feature descriptions for variables to be generated, with corresponding values $x_1, \ldots, x_d$, support set S, semantic context c and predictive context p_{pc}, then $p_\gamma(x_1, \ldots, x_d \mid f_1, \ldots, f_d, S, c, p_{\mathrm{pc}}) = \prod_{t=1}^{d} p_\gamma\big(x_t \mid \{(f_m, x_m)\}_{m<t}, f_t, S, c, p_{\mathrm{pc}}\big)$. That is, ASPIRE returns native conditional distributions for arbitrary targets, these factors can be queried autoregressively to perform zero-shot or few-shot generation-style inference/generation over variables. In this sense, the model provides a native conditional inference interface that extends beyond ordinary point prediction.

Table 1: Representative tabular foundation models across capabilities relevant to zero/few-shot arbitrary conditioning (✱ denotes partial support; here this means that the method captures part of the intended capability but not the stricter version used in our arbitrary conditioning framing, as discussed in the text).

Method	PIV Features	PIV Shots	Semantic Context	Native Cont. pdf	Zero-shot Inference	In-Context Learning
TP-BERTa Yan et al. (2024)	✓	✗	✓	✗	✗	✗
CM2 Ye et al. (2024)	✓	✗	✓	✗	✓	✗
GTL Wen et al. (2024)	✗	✗	✓	✗	✓	✓
TabSTAR Arazi et al. (2025)	✓	✗	✓	✗	✗	✗
TabPFNv2.5 Hollmann et al. (2025)	✱	✓	✗	✱	✗	✓
TabICL QU et al. (2025)	✗	✓	✗	✗	✗	✓
LimiX Zhang et al. (2025)	✗	✓	✗	✱	✗	✓
ConTextTab Spinaci et al. (2025)	✱	✓	✓	✗	✗	✓
TabFM Research (2026)	✱	✓	✗	✗	✗[†]	✓
ASPIRE	✓	✓	✓	✓	✓	✓

[†]Zero-shot in TabFM is defined as inference with no weight updates; we define it as $k{=}0$ support rows.

3 Related Works

ASPIRE is motivated by the observation that predicting arbitrary targets from arbitrary observed subsets across heterogeneous datasets requires more than either cross-table transfer or masked prediction alone: the model must generalize simultaneously across semantic context, target variables, and observation patterns. Several key characteristics are core to arbitrary conditioning over zero/few-shots: *PIV Features* and *PIV Shots*—invariance to feature and support-example order respects the structure of the problem for better representation learning; *Semantic Context*—semantic contexts across heterogeneous schemas grounds the prediction task when little or no support instances are available; *Native Continuous pdfs*—continuous targets should be modeled in their native domain, rather than only through discretized labels or point estimates, to support calibrated likelihoods and uncertainty-aware reasoning; *Zero-shot/ Few-shot In-Context Learning*—optional support-based in-context inference implies that the model learns to ingest support examples to better center its predictive context. Existing methods capture some of these capabilities individually, but ASPIRE integrates all of them jointly within a single arbitrary-conditioning interface (Tab. 1).

Tabular Foundation Models Recent tabular foundation models have substantially advanced cross-dataset learning, but they differ in which parts of the arbitrary conditioning problem they address. Early cross-table methods such as XTab (Zhu et al., 2023), CARTE (Kim et al., 2024), TransTab (Wang & Sun, 2022), TaBERT (Yin et al., 2020), TP-BERTa (Yan et al., 2024), and CM2 (Ye et al., 2024) focus primarily on learning transferable representations that can be adapted to downstream tasks on new datasets. Serialization-based approaches such as TabLLM (Hegselmann et al., 2023) and GTL (Wen et al., 2024) instead cast tabular prediction as sequence modeling over row-text prompts, while FeatLLM (Han et al., 2024) uses an LLM to generate derived features before fitting a separate predictor. Among these, CM2 and GTL are the closest precedents to ASPIRE: CM2 because it already uses masking-based target-from-context prediction with semantic schema information, and GTL because it directly targets zero-shot and in-context tabular inference through semantic prompting. However, neither method is organized around the architectural decomposition used by ASPIRE. More broadly, these methods are not built around the same target-native conditional likelihood interface over arbitrary targets and observed subsets that ASPIRE exposes.

A complementary line of work is represented by TabPFN family including TabPFNv2/v3 (Hollmann et al., 2022; 2025; Grinsztajn et al., 2026),TabFM (Research, 2026), which casts tabular in-context learning as approximate posterior predictive inference over labeled support examples. While subsequent methods extend it through different priors and architectures: TabForestPFN (den Breejen et al., 2025) incorporates tree-based priors, TabDPT (Ma et al., 2025) introduces real-data pretraining, TabICL (QU et al., 2025) uses column-then-row attention for efficient in-context learning, and LimiX (Zhang et al., 2025) broadens conditional modeling through joint treatment of feature values and missingness. ConTextTab (Spinaci et al., 2025) injects semantic alignment into the ICL

framework via modality-specific embeddings trained on large-scale real tables. These methods are primarily designed to infer a designated target from labeled in-context examples, whereas ASPIRE treats the target feature and observed subset themselves as part of the conditional query and supports inference with either no support examples or only a very small support set. They also differ in several aspects of the inference interface emphasized here: TabPFNv2 is only partially permutation-aligned because its random feature embeddings break strict feature-order invariance (Hollmann et al., 2025; Ye et al., 2025; QU et al., 2025); TabPFN's regression is distributional but discretized; and LimiX, despite its broader conditional-query framing, does not clearly specify a comparable native continuous density beyond a regression head.

Arbitrary Conditional Distribution Arbitrary conditioning—predicting arbitrary targets from arbitrary observed subsets—is a core primitive for probabilistic inference. Prior work typically studies this problem within a fixed feature universe, often for a single dataset. The Universal Marginalizer (Douglas et al., 2017) predicts missing variables from observed subsets, while VAEAC (Ivanov et al., 2018), Neural Conditioner (Belghazi et al., 2019), ACFlow (Li et al., 2020), and ACE (Strauss & Oliva, 2021) model increasingly expressive arbitrary conditional distributions. These methods primarily generalize across missingness patterns within a fixed schema. ASPIRE instead extends arbitrary conditioning across heterogeneous datasets by grounding targets and observed features in semantic descriptions for both zero-shot and few-shot inference with a small optional support set.

4 Experiments

We train ASPIRE on **1,400** diverse tabular datasets from OpenTabs (Ye et al., 2024), each annotated with natural-language dataset and feature descriptions. We use the held-out evaluation split from OpenTabs Ye et al. (2024) for 15 classification and 5 regression tasks, and extend the suite with additional 40 held out regression tasks from CARTE (Kim et al., 2024). Following Spinaci et al. (2025), we conduct a contamination study to verify that held-out evaluation tasks do not overlap with the training corpus at the dataset, schema, feature-description, or paraphrase level. In total, datasets span healthcare, finance, and scientific domains, with 10–500 features and 1K–1M samples. This evaluation scale is consistent with recent cross-table learning studies, including CARTE and TabSTAR, which evaluate transfer on held-out suites of comparable size (Kim et al., 2024; Arazi et al., 2025). ASPIRE uses BERT (Devlin et al., 2019) for text encoding and an 8-layer Set Transformer backbone for set-based representation learning. We report F1 for classification and RMSE for regression, averaged over three random seeds. Continuous targets normalized so RMSE is comparable across tasks with different target scales. During training, we augment dataset- and feature-level descriptions with LLM-generated paraphrases to improve linguistic robustness. Baseline citations appear in Sec. 3. See App. A.4 for preprocessing and hyperparameters; open-source code will be released upon publication.

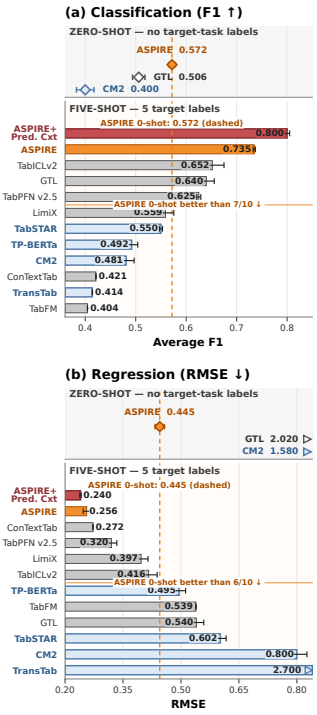

Figure 3: Zero-/few-shot performance on held-out classification and regression tasks.

We align input representations so that baselines receive the same information as ASPIRE wherever their architectures permit. Baselines highlighted in blue in Fig. 3 are trained from scratch on our OpenTabs training corpus, using feature and dataset descriptions when their architectures support semantic inputs. For TabSTAR, TP-BERTa, TransTab, and CM2, few-shot adaptation follows the CM2 protocol of Ye et al. (2024): the pretrained model is fine-tuned on the k sampled support rows. The remaining methods use in-context support examples, and we average results across support-set draws.

TabFM, ConTextTab, TabPFNv2.5, TabICLv2, and LimiX are evaluated using their released pretrained checkpoints and recommended protocols. We also evaluated TabPFNv3, but omit its results from the figure because they were slightly worse or comparable to those of TabPFNv2.5 (see Tab. 3). GTL follows its recommended pretrained-LLM protocol because retraining it on OpenTabs is

computationally prohibitive. These checkpoints are not retrained by us on OpenTabs: they retain their respective pretraining regimes, which often use large collections of synthetic tabular tasks or a separate corpus of real tables.

Figure 3 compares direct inference on held-out schemas with zero or five target-dataset examples. In the zero-shot setting, ASPIRE achieves an average F1 of 0.572, compared with 0.506 for GTL and 0.400 for CM2. Its regression RMSE is 0.445, compared with 1.580 and 2.020, respectively. Even without target-dataset labels, ASPIRE exceeds the five-shot results of 7 of the 10 plotted classification baselines and 6 of the 10 regression baselines. These results show that ASPIRE's semantic context integration and training support useful conditional predictions before examples from the new dataset are available.

With five support examples, ASPIRE improves to 0.735 F1 and 0.256 RMSE. The strongest plotted alternatives achieve 0.652 F1 (TabICL v2) and 0.272 RMSE (ConTextTab), corresponding to 12.7% higher F1 and 5.9% lower RMSE for ASPIRE. Incorporating TabPFNv3's forecast through ASPIRE's predictive-context and mixing mechanisms further improves performance to 0.800 F1 and 0.240 RMSE. (We examine the respective contributions of forecast conditioning and mixing below.) We further note that full dataset transfer results follow similar trends (see Appendix Fig. 4)

The performance across target types is also notable. Among the 10 five-shot baselines, ConTextTab is the strongest competing regressor (0.272 RMSE) but ranks eighth in classification (0.421 F1), below ASPIRE's zero-shot F1. Conversely, TabICLv2 is the strongest competing classifier (0.652 F1) but ranks fourth in regression (0.416 RMSE), only slightly ahead of ASPIRE's zero-shot RMSE. ASPIRE leads the five-shot comparisons on both metrics, with gains spanning classification and regression.

4.1 Ablation Studies

We systematically ablate ASPIRE's key components in Table 2, yielding four main observations. **Likelihood-based training is crucial for regression**: replacing ASPIRE's mixture-of-Gaussians likelihood objective with a point-estimate MSE loss substantially worsens regression ($0.256 \rightarrow 0.348$ RMSE), highlighting the importance of target-native likelihoods for continuous targets. **Semantic grounding improves transfer**: removing feature descriptions reduces performance from 0.735 to 0.696 F1 and increases RMSE from 0.256 to 0.275, while removing dataset descriptions yields 0.699 F1 and 0.260 RMSE. This indicates that both dataset- and feature-level semantics are useful, with feature descriptions contributing more strongly in this setting. **Intra-instance set reasoning matters**: removing IntraSet2Set reduces F1 from 0.735 to 0.690 and increases RMSE from 0.256 to 0.265, showing the value of modeling co-occurring feature–value dependencies before aggregation. **Permutation invariance is critical**: replacing ASPIRE's role/type embeddings with positional encodings breaks the invariance described in §2.2, reducing F1 from 0.735 to 0.643 and increasing RMSE from 0.256 to 0.267. Together, these results show that ASPIRE's gains come from likelihood-consistent training, semantic grounding, permutation-equivariant intra-instance reasoning, and permutation-invariant aggregation.

Table 2: ASPIRE Ablation. 5-shot learning F1 and RMSE averaged across test datasets.

Variant	F1 ↑	RMSE ↓
Full ASPIRE	0.735	0.256
Training Objective		
MSE Loss	0.725	0.348
Semantic Grounding		
w/o Feat Desc.	0.696	0.275
w/o Dataset Desc.	0.699	0.260
Permutation Equivariance		
w/o IntraSet2Set	0.690	0.265
Permutation Invariance		
Positional Enc.	0.643	0.267

Ablating Predictive Context and Ensembling. We vary whether ASPIRE receives an external forecast as predictive context during training and inference, and how its output is combined with that forecast (Tab. 3). The external model is TabPFNv3, which receives the same five support examples available to ASPIRE. We compare ASPIRE with and without TabPFNv3's forecast as input ("Pred. context"), under either equal-weight soft voting (SV) (Maillet et al., 2026) or our query-specific dynamic mixing (Dyn.). We also report both models alone and *Greedy Ensemble Selection* (GES) (Caruana et al., 2004). GES selects mixing weights using leave-one-out validation on the five support examples: each example is evaluated in turn using the other four as support.

Table 3: Predictive context ablation for 5-shot learning F1 and RMSE averaged across test datasets

	F1 ↑		RMSE ↓	
Pred. context	SV	Dyn.	SV	Dyn.
Yes	0.787	**0.800**	0.245	**0.240**
No	0.760	0.771	0.252	0.249
GES ensembling	0.654		0.353	
TabPFNv3 alone	0.630		0.318	
ASPIRE alone	0.735		0.256	

The results in Table 3 show a gain beyond the "sum of the parts": even with the same 50/50 mixture, performance improves when ASPIRE ingests TabPFNv3's forecast (Yes/SV versus No/SV). The resulting predictions also outperform TabPFNv3 alone, rather than merely reproducing its forecast. Dynamic mixing provides further gains both with and without predictive context, with our full approach (Yes/Dyn.) yielding the best results.

4.2 Generative Quality via Chain Rule

We next move beyond supervised prediction to evaluate whether the models can generate synthetic tables that retain statistical structure when their conditional predictions are composed over multiple steps. For each held-out test dataset, we synthesize a table with the same number of rows as the real table, using a five-shot support set for ASPIRE and the predictive baselines. Each row is generated autoregressively in a random feature order, conditioning each sampled feature on the values generated earlier in that row. Further implementation details for each baseline are provided in the App. A.5.1

Table 4: Distributional fidelity on held-out datasets. Higher WNMIS is better; C2ST AUC closer to 0.5 is better.

Method	WNMIS ↑	C2ST AUC ↓
ASPIRE	**0.9780**	**0.5400**
CM2	0.9210	0.7200
TabPFNv2.5	0.9560	<u>0.5540</u>
LimiX	0.9020	0.5640
TabICL v2	<u>0.9710</u>	0.5880
TP-BERTa	0.9350	0.7010
TabPFNv3	0.9426	0.6880
ConTextTab	0.8098	0.9681
TabFM	0.9369	0.8037
CLLM	0.7877	0.9959
StructSynth	0.7350	0.9990

As noted by Scassola et al. (2026), many metrics can fail to accurately capture generative data quality in tabular settings. We thus follow their suggestions and jointly evaluate distributional fidelity with weighted normalized mutual information similarity (WNMIS) and a classifier two-sample test (C2ST AUC). WNMIS measures agreement in pairwise dependencies between real and synthetic data, while C2ST captures broader distributional differences by training an XGBoost discriminator to distinguish real from synthetic instances. Additional column-wise metrics (KS for numerical features and TVD for categorical features) following Zhang et al. (2024) are reported in the Appendix and show similar trends.

ASPIRE achieves the strongest five-shot generation performance, attaining the highest WNMIS and the C2ST AUC closest to 0.5 among the evaluated baselines. These include semantically grounded models, tabular foundation models, and few-shot LLM-based synthesis methods such as CLLM (Seedat et al., 2024) and StructSynth (Liu et al., 2025). Improvements over the strongest competing baselines are statistically significant using a paired t-test for both WNMIS ($p = 0.007$) and C2ST AUC ($p = 0.005$). ASPIRE leads on both metrics, whereas the runner-up differs: TabICLv2 for WNMIS and TabPFNv2.5 for C2ST AUC.

5 Discussion

Zero- and few-shot arbitrary conditioning across unseen schemas requires a model to interpret changing targets and observed feature subsets while incorporating support examples and external predictions when available. ASPIRE addresses this with semantically grounded feature–value atoms, permutation-equivariant reasoning within instances, and role-aware integration of evidence. Its target-native conditional likelihoods support classification, regression, and chain-rule generation through repeated conditional sampling.

Limitations and Future Work. As with much of state-of-the-art cross-dataset learning, ASPIRE focuses on tabular data and relies on opaque neural inference. Future work should extend arbitrary conditioning to additional modalities, such as images, sequences, graphs, and text, and incorporate more interpretable modeling to improve knowledge discovery while preserving flexible arbitrary conditional inference.

Results and Conclusion. Empirically, ASPIRE performs strongly across our evaluation suite, spanning held-out classification, regression, conditional inference, synthetic generation. More broadly, these results suggest that *arbitrary conditioning, structured set-based reasoning, and target-native conditional likelihood modeling, and in-context ensembling are useful design principles for cross-tabular learning*.

6 AI USE STATEMENT

In this work, we used generative AI tools to check for grammatical errors and language editing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

7 ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our study is methodological and empirical and does not involve recruiting human subjects or collecting new personal data. The experiments use existing research datasets according to their intended research use and applicable terms and licenses.

8 REPRODUCIBILITY STATEMENT

We provide implementation and experimental details throughout the paper and appendix to support reproducibility. We also attach code in the supplementary material and will be open-sourced upon publication.

REFERENCES

Alan Arazi, Eilam Shapira, and Roi Reichart. Tabstar: A tabular foundation model for tabular data with text fields, 2025. URL https://arxiv.org/abs/2505.18125.

Mohamed Belghazi, Maxime Oquab, and David Lopez-Paz. Learning about an exponential amount of conditional distributions. *Advances in Neural Information Processing Systems*, 32, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. 07 2004. doi: 10.1145/1015330.1015432.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of machine learning research*, 25(70):1–53, 2024.

Edo Cohen-Karlik, Avichai Ben David, and Amir Globerson. Regularizing towards permutation invariance in recurrent models. *Advances in Neural Information Processing Systems*, 33:18364–18374, 2020.

Felix den Breejen, Sangmin Bae, Stephen Cha, and Se-Young Yun. Fine-tuned in-context learning transformers are excellent tabular data classifiers, 2025. URL https://arxiv.org/abs/2405.13396.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Laura Douglas, Iliyan Zarov, Konstantinos Gourgoulias, Chris Lucas, Chris Hart, Adam Baker, Maneesh Sahani, Yura Perov, and Saurabh Johri. A universal marginalizer for amortized inference in generative models. *arXiv preprint arXiv:1711.00695*, 2017.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller,

Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-3: Technical report, 2026. URL https://arxiv.org/abs/2605.13986.

Sungwon Han, Jinsung Yoon, Sercan O Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning, 2024. URL https://arxiv.org/abs/2404.09491.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, pp. 5549–5581. PMLR, 2023.

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.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. Carte: pretraining and transfer for tabular learning. *arXiv preprint arXiv:2402.16785*, 2024.

Masanari Kimura, Ryotaro Shimizu, Yuki Hirakawa, Ryosuke Goto, and Yuki Saito. On permutation-invariant neural networks. *arXiv preprint arXiv:2403.17410*, 2024.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Yang Li, Shoaib Akbar, and Junier Oliva. Acflow: Flow models for arbitrary conditional likelihoods. In *International Conference on Machine Learning*, pp. 5831–5841. PMLR, 2020.

Siyi Liu, Yujia Zheng, and Yongqi Zhang. Structsynth: Leveraging llms for structure-aware tabular data synthesis in low-data regimes, 2025. URL https://arxiv.org/abs/2508.02601.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L Caterini, and Maksims Volkovs. Tabdpt: Scaling tabular foundation models on real data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Vassili Maillet, Gustavo, Angulo, and Pierre Jouvelot. Choosing a parallel heterogeneous ensemble method for tabular classification, 2026. URL https://arxiv.org/abs/2607.05103.

Jingang QU, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0VvD1PmNzM.

Google Research. Tabfm: A zero-shot foundation model for tabular data. 2026. URL https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Davide Scassola, Dylan Ponsford, Adrián Javaloy, Sebastiano Saccani, Luca Bortolussi, Henry Gouk, and Antonio Vergari. A sobering look at tabular data generation via probabilistic circuits. In *The Ninth Workshop on Tractable Probabilistic Models*, 2026. URL https://openreview.net/forum?id=HoArXIgROj.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. Curated llm: Synergy of llms and data curation for tabular augmentation in low-data regimes, 2024. URL https://arxiv.org/abs/2312.12112.

Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. Contexttab: A semantics-aware tabular in-context learner. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kGMRb4jbTP.

Ryan Strauss and Junier B Oliva. Arbitrary conditional distributions with energy. *Advances in Neural Information Processing Systems*, 34:752–763, 2021.

Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A Osborne. On the limitations of representing functions on sets. In *International conference on machine learning*, pp. 6487–6494. PMLR, 2019.

Peihao Wang, Shenghao Yang, Shu Li, Zhangyang Wang, and Pan Li. Polynomial width is sufficient for set representation with high-dimensional features. *arXiv preprint arXiv:2307.04001*, 2023.

Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. In *Advances in Neural Information Processing Systems*, 2022.

Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3323–3333, 2024.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction. *arXiv preprint arXiv:2403.01841*, 2024.

Chao Ye, Guoshan Lu, Haobo Wang, Liyao Li, Sai Wu, Gang Chen, and Junbo Zhao. Towards cross-table masked pretraining for web data mining. In *Proceedings of the ACM Web Conference 2024*, pp. 4449–4459, 2024.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabpfn v2: Strength, limitation, and extension. *arXiv preprint arXiv:2502.17361*, 2025.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space, 2024. URL https://arxiv.org/abs/2310.09656.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, et al. Limix: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025.

Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. Xtab: Cross-table pretraining for tabular transformers. *arXiv preprint arXiv:2305.06090*, 2023.

A APPENDIX

A.1 FULL DATASET TRANSFER

We next evaluate models' ability to transfer under supervised adaptation when ample training data are available. Figures 4 report many-shot transfer results for classification and regression, respectively. Each model is fine-tuned on the target dataset's training split using AdamW, with early stopping on a held-out validation set, and evaluated on a fixed test split under a consistent 70%/15%/15% train/validation/test protocol. In-context–learning-based models are fine-tuned following the procedure of Hollmann et al. (2025); Across datasets, ASPIRE achieves the strongest performance, indicating that its learned representations provide an effective initialization for supervised transfer and full-data adaptation. Taken together, these results underscore ASPIRE's versatility: the same model supports both in-context few-shot inference—consistent with zero- and few-shot arbitrary conditioning objectives—and end-to-end supervised fine-tuning.

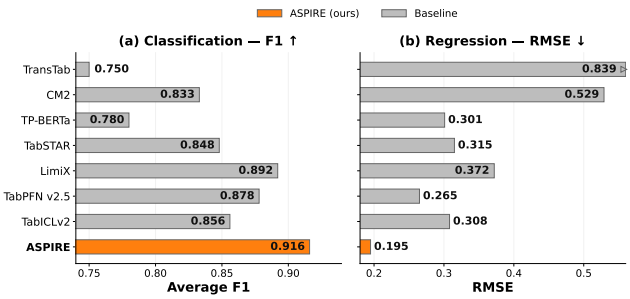

Figure 4: Fine-tuning comparison on classification and regression tasks.

A.2 ADDITIONAL RELATED WORKS

A.2.1 SET MODELING

Early methods enforce permutation invariance by augmenting training data with randomly permuted versions of input sets, treating them as sequences while training models to produce consistent outputs across permutations. However, this approach does not guarantee invariance in practice—especially with finite data and limited model capacity—since sequence models inherently exploit positional information (Zaheer et al., 2017).

A foundational work in this area is DeepSets (Zaheer et al., 2017), which proves that any continuous permutation invariant function can be expressed as $f(S) = h\left(\sum_{x \in S} g(x)\right)$, where g maps individual elements and h aggregates the result. This leads to a simple yet expressive two-stage neural architecture. DeepSets also introduces permutation equivariant layers through shared transformations and pooling operations to capture intra-set dependencies. However, later work (Wagstaff et al., 2019) shows that the latent dimension of such architectures must grow at least linearly with the set size to maintain universal approximability, which may limit their practicality.

To model richer interactions between set elements, Set Transformer (Lee et al., 2019) replaces pooling with self-attention mechanisms. Attention layers are inherently permutation equivariant as they compute weighted sums over all elements. By combining these with attention-based pooling, Set Transformer yields permutation-invariant representations while capturing complex intra-set relationships—an approach that lays the foundation to our ASPIRE architecture. Several extensions have emerged to address specific limitations. Hölder-based power means, and quasi-arithmetic pooling strategies (Kimura et al., 2024) generalize sum and max pooling for increased expressivity. Other approaches like subset-invariant regularization (Cohen-Karlik et al., 2020) enforce permutation symmetry via learning objectives rather than architectural constraints. Recent work (Wang et al., 2023) provides refined insights into the trade-offs between model width, depth, and set size for maintaining expressivity.

A.3 DATA PROCESSING

A.3.1 TRAINING DATA

We use datasets from OpenTabs (Ye et al., 2024) for training, validation, and testing. We manually collect dataset descriptions and feature descriptions from UCI ML repository, Kaggle, OpenML, etc.

Table 5: Downstream datasets used in our experiments.

Dataset Name	R/C	Samples	Numerical	Categorical	Label Classes	Source
Breast	C	699	9	0	2	https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original
Bone	C	1479	2	7	3	https://archive.ics.uci.edu/dataset/3/connectionist+bench+choice
Diabetes	C	768	8	0	2	https://openml.org/d/37
Vehicle	C	846	18	0	4	https://archive.ics.uci.edu/dataset/149/statlog+vehicle+silhouettes
Satimage	C	6430	36	0	6	https://archive.ics.uci.edu/dataset/146/statlog+landsat+satellite
Sick	C	3772	7	22	2	http://archive.ics.uci.edu/dataset/102/thyroid+disease
Analcatdata	C	797	0	4	6	https://pages.stern.nyu.edu/jsimonof/AnalCatData/Data/
Pcl	C	1109	21	0	2	https://openml.org/d/1068
Adult	C	48842	6	8	2	https://archive.ics.uci.edu/dataset/2/adult
PhishingWebsites	C	11055	0	30	2	https://archive.ics.uci.edu/dataset/327/phishing+websites
Cylinder-bands	C	540	18	21	2	https://archive.ics.uci.edu/dataset/32/cylinder+bands
MiceProtein	C	1080	77	4	8	https://archive.ics.uci.edu/dataset/342/mice+protein+expression
Car	C	1728	0	6	4	https://archive.ics.uci.edu/dataset/19/car+evaluation
Segment	C	2310	19	0	7	http://archive.ics.uci.edu/dataset/50/image+segmentation
Porto-seguro	R	2000	26	31	2	https://openml.org/d/44787
Amazon	C	2000	0	9	2	https://openml.org/d/44712
Elevators	R	16599	18	19	–	https://openml.org/d/216
Yprop	R	8885	251	0	–	https://openml.org/d/416
Topo	R	8885	266	267	–	https://openml.org/d/422
SAT11	R	4400	115	1	–	https://www.cs.ubc.ca/labs/algorithms/Projects/SATzilla/
Diamonds	R	53940	6	3	–	https://openml.org/d/42225
House_sales	R	21613	20	1	–	https://openml.org/d/42731

Further, we curate metadata about the dataset by obtaining statistics about the dataset using python functions, for example, collecting potential classes for each target, data type of the feature values etc. Tables which have too few rows and columns, which have unclear and invalid data, are dropped. We will open-source this metadata upon publication. We identify task as classification or regression based on target value types, followed by standarization of continuous values. Feature descriptions are embedded using pre-trained transformer models (BERT-base-uncased) to create dense semantic representations that capture feature semantics across diverse domains.

A.3.2 Evaluation Data

The following table 5 from OpenTabs (Ye et al., 2024) and regression tasks from the CARTE benchmark suite (Kim et al., 2024) details information about the downstream datasets we use as test datasets in our evaluation.

A.4 ASPIRE Architecture

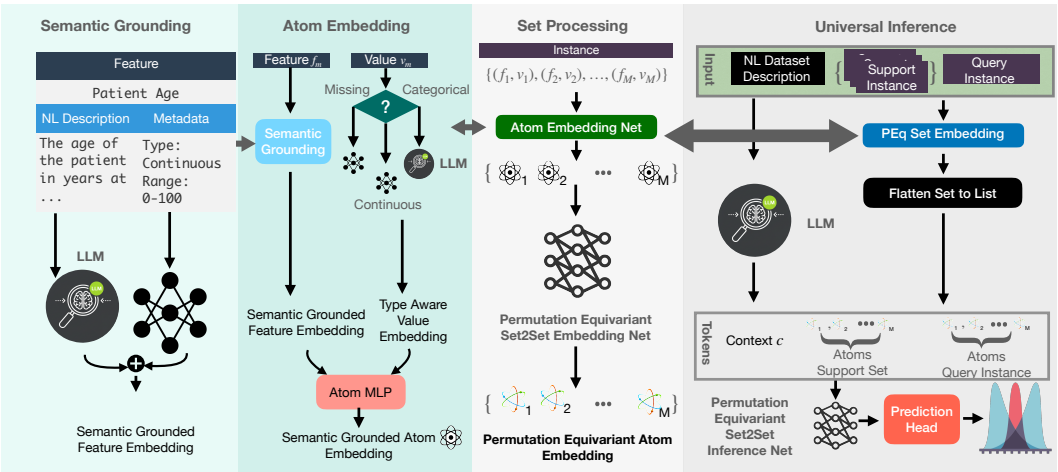

Figure 5: ASPIRE processing pipeline. The framework transforms heterogeneous inputs from different domains through semantic alignment into a shared understanding space, applies permutation-invariant set processing to capture feature interactions while maintaining order independence, and produces universal predictions that support arbitrary conditioning and cross-domain transfer.

In addition to ASPIRE model descriptions in the paper, the following implementation details are relevant. Our optimization uses AdamW with a learning rate of 1e-4 and cosine annealing with warm restarts (100 warmup steps, weight decay of 0.01-0.04). During training, we apply feature

dropout with a 40% masking rate, where selected features have their values replaced with missing value embeddings to improve generalization across diverse tabular domains. Feature encoding uses specialized approaches: real-valued features utilize Fourier embeddings with 256 learnable frequency components spanning 1-256, while categorical features undergo BERT encoding followed by projection to the model dimension of 768. The Set Transformer components use 4-8 attention heads with 16-32 induced points for permutation-invariant row representations.

The multi-task learning framework combines cross-entropy loss for classification and mixture-of-Gaussians negative log-likelihood for regression, where the regression head employs 10 Gaussian components with learnable weights, means, and log-variances. Training incorporates early stopping with patience of 5-7 epochs. High level pipeline is shown in Fig. 5

A.5 Details on Synthetic Data Generation

A.5.1 Chain Rule Implementation

For each synthetic row, we independently sample a different random feature ordering and generate the features sequentially according to that ordering. At step t, a semantic model like ASPIRE, CM2, TP-BERTa, or ConTextTab receives the dataset and feature descriptions, a randomly sampled five-row support set, the feature currently being generated, and only the values generated at earlier steps. Models without a semantic interface like TabPFNv2.5, TabICLv2, TabFM instead receive the corresponding column identities, feature types, support values, and generated prefix in their standard tabular representation. Ungenerated features remain masked. For the first feature, the generated prefix is empty, and the distribution is estimated using only the support set rather than the full-data empirical marginal. Categorical values are sampled from predicted class probabilities, while continuous values are sampled from the model's predictive density. This process is repeated until the synthetic table contains the same number of rows as the real table.

A.5.2 Distributional Fidelity Metrics

We evaluate the fidelity of generated tables by comparing each synthetic table to the corresponding held-out real table at the column-marginal level as mentioned in (Zhang et al., 2024). For each dataset, every method generates a synthetic table with the same number of rows as the real test table. We then compute feature-wise distributional distances between real and synthetic columns, using Total Variation Distance (TVD) for categorical features and the Kolmogorov–Smirnov (KS) statistic for continuous features. Lower values indicate better distributional fidelity.

For a categorical feature j, let P_j and $\widehat{P}_j$ denote the empirical category distributions in the real and synthetic tables, respectively. We compute TVD as

$$\mathrm{TVD}_j(P_j, \widehat{P}_j) = \frac{1}{2} \sum_{c \in \mathcal{C}_j} \left| P_j(c) - \widehat{P}_j(c) \right|,$$

where $\mathcal{C}_j$ is the union of categories appearing in either the real or synthetic column. This metric is bounded in $[0, 1]$, with 0 indicating identical empirical categorical distributions.

For a continuous feature j, let F_j and $\widehat{F}_j$ denote the empirical cumulative distribution functions of the real and synthetic values. We compute the two-sample KS statistic as

$$\mathrm{KS}_j(F_j, \widehat{F}_j) = \sup_x \left| F_j(x) - \widehat{F}_j(x) \right|.$$

Unlike histogram-based distances, the KS statistic does not require discretizing continuous values and directly measures the maximum vertical gap between the two empirical CDFs.

For each dataset d, we aggregate feature-level distances by averaging over the corresponding feature types:

$$\mathrm{TVD}^{(d)} = \frac{1}{|\mathcal{C}^{(d)}|} \sum_{j \in \mathcal{C}^{(d)}} \mathrm{TVD}_j, \qquad \mathrm{KS}^{(d)} = \frac{1}{|\mathcal{N}^{(d)}|} \sum_{j \in \mathcal{N}^{(d)}} \mathrm{KS}_j,$$

where $\mathcal{C}^{(d)}$ and $\mathcal{N}^{(d)}$ denote the categorical and numerical feature sets of dataset d, respectively. We then report the mean TVD and KS across held-out datasets, averaging over random seeds when

Table 6: Generative ability on held-out test datasets.

Model	TVD ↓	KS ↓
5-shot		
ASPIRE	**0.1506**	**0.1880**
CM2	0.2961	0.4662
TPFNv2.5	0.2478	0.2026
LimiX	0.3470	0.6422
TabICLv2	0.1669	0.1982
TP-BERTa	0.4176	0.6393
CLLM	0.3935	0.4291
StructSynth	0.4248	0.5994

multiple generations are produced. Missing, non-finite, or invalid generated values are removed before metric computation, and feature types are kept fixed across real and synthetic tables. These metrics evaluate whether repeated conditional generation preserves the marginal distributions of mixed-type tabular features.

Weighted Normalized Mutual Information Similarity. WNMIS compares pairwise normalized mutual information (NMI) between the real and synthetic tables. For two features x_i and x_j, normalized mutual information is

$$\mathrm{NMI}(x_i, x_j) = \frac{2I(x_i; x_j)}{H(x_i) + H(x_j)} \in [0, 1],$$

where $I(\cdot;\cdot)$ denotes mutual information and $H(\cdot)$ denotes entropy. For continuous features, the variables are discretized before estimating mutual information, following Scassola et al. (2026).

Let $\mathrm{NMI}_R(x_i, x_j)$ and $\mathrm{NMI}_S(x_i, x_j)$ denote the NMI computed from the real and synthetic tables, respectively. Pairwise similarity is defined as

$$\mathrm{NMIS}(x_i, x_j) = 1 - \left|\mathrm{NMI}_R(x_i, x_j) - \mathrm{NMI}_S(x_i, x_j)\right|.$$

Pairs are weighted according to the strength of their dependency in the two tables,

$$w(x_i, x_j) = \left|\mathrm{NMI}_R(x_i, x_j) + \mathrm{NMI}_S(x_i, x_j)\right|, \qquad \widetilde{w}(x_i, x_j) = \frac{w(x_i, x_j)}{\sum_{a<b} w(x_a, x_b)}.$$

The resulting WNMIS score for a dataset is

$$\mathrm{WNMIS} = \sum_{i<j} \widetilde{w}(x_i, x_j)\, \mathrm{NMIS}(x_i, x_j).$$

WNMIS lies in $[0, 1]$, with larger values indicating stronger agreement between the pairwise dependency structures of the real and synthetic tables. Unlike metrics based only on correlations or marginal distributions, WNMIS uses mutual information to capture potentially nonlinear associations and is designed to be less sensitive to marginal fidelity alone.

Classifier Two-Sample Test. C2ST evaluates joint-distribution fidelity by training a binary classifier to distinguish real from synthetic rows. Given equally sized real and synthetic datasets $\mathcal{D}_R$ and $\mathcal{D}_S$, we train a classifier c on their training splits and evaluate it on held-out rows. Following Scassola et al. (2026), we report the C2ST-AUC:

$$\text{C2ST-AUC}(c, \mathcal{D}_R, \mathcal{D}_S) = \mathrm{AUROC}\Big(c; \mathcal{D}_R^{(\mathrm{test})}, \mathcal{D}_S^{(\mathrm{test})}\Big).$$

A value close to 0.5 indicates that real and synthetic rows are indistinguishable and therefore corresponds to better fidelity, whereas a value close to 1 indicates that they are easily distinguishable.

A.6 Baseline Details

CM2: We use the pre-trained model, which have number of transformer layers 3, attnetion heads 8, batch size 256, learning rate for finetuning 3e-4, patience is 5. For performing few-shot learning, Ye et al. (2024) prescribes k-shot learning via finetuning the model with k examples sampled from the

train set. We observe that this causes high-variance in the performance, therefore we report the mean of all metrics for CM2 few-shot learning experiments.

GTL: We use the authors' released GTL model (a LLaMA-2 backbone; 7B/13B parameters depending on the checkpoint) and the official inference pipeline. Unless otherwise stated, we keep the default evaluation settings provided by the authors and do not perform dataset-specific hyperparameter tuning. When we run k-shot adaptation for GTL, we follow the training configuration reported in the GTL work (AdamW optimizer, learning rate 1×10^{-5}, batch size 512, max sequence length 4096, and 256 update steps).

LimiX: We evaluate the publicly released LimiX checkpoints (LimiX-16M, and LimiX-2M when reporting the lightweight variant). LimiX is used in its training-free, context-driven inference mode: we provide the labeled training split as context and query the model for predictions on the test split, without any dataset-specific fine-tuning. We run the official predictor/inference interface and use the default inference configuration from the release (enabling mixed precision when available).

TabICLv2: We use the pre-trained TabICLv2 checkpoint and the official inference code. TabICLv2 performs dataset-wise in-context learning by conditioning on the labeled training set and predicting the test set in a single forward pass without parameter updates. For large tables, we retain the authors' default memory-management behavior, including automatic batch-size selection based on available GPU memory and optional CPU/disk offloading.

TabPFNv2.5/v3: We use the official TabPFN-2.5 release via the authors' provided implementation, and evaluate it in the standard PFN in-context regime (training split as context; no parameter updates). We tune hyperparameters per dataset for prediction.

TabFM: We use the official release on HuggingFace and evaluate in the same in-context learning setting as TabPFNv3.

ConTextTab: We use the official sap-rpt-1-oss release via the authors' provided implementation and evaluate it in the standard tabular in-context learning regime. For datasets where semantic feature information is available, we provide the same feature descriptions used by all other methods capable of leveraging semantic metadata.

TransTab: We train TransTab from scratch using their provided training code on the heterogeneous OpenTabs table pool.

TP-BERTa: We add TP-BERTa as an additional LM-based tabular baseline. TP-BERTa is initialized from a RoBERTa backbone adapted for tabular prediction and is designed for standard supervised tabular prediction via fine-tuning on downstream datasets. Following the authors' guidance, we fine-tune TP-BERTa per dataset with a larger training budget (up to 200 epochs) and early stopping with patience 50 epochs.

Table 7: **Replication check for LimiX-16M on the BCCO benchmark.** We reproduce the BCCO classification (AUC/ACC/F1; higher is better) and regression (RMSE; lower is better, and R^2; higher is better) results for LimiX-16M. We report our self-run averages, the paper-reported averages, and Δ (self-run $-$ paper).

	AUC ↑			ACC ↑			F1 ↑		
Setting	**Self**	**Paper**	**Δ**	**Self**	**Paper**	**Δ**	**Self**	**Paper**	**Δ**
BCCO-CLS	86.02	87.10	-1.08	78.60	80.40	-1.79	69.77	73.10	-3.33
	RMSE ↓			R^2 ↑					
Setting	**Self**	**Paper**	**Δ**	**Self**	**Paper**	**Δ**			
BCCO-REG	38.78	38.60	0.18	78.99	79.40	-0.41			

Table 8: **Replication check for GTL on unseen Kaggle holdout tasks.** We report self-run performance on the GTL holdout split alongside paper-reported values and Δ (self-run $-$ paper). Results are averaged over 20 classification and 24 regression tasks.

Setting	Self-run	Paper	Δ
Classification (avg.)	0.684	0.714	-0.030
Regression (avg.)	0. 301	0.288	0.013