JoLT: Joint Probabilistic Predictions on Tabular Data Using LLMs

Aliaksandra Shysheya[*]
AS2975@CAM.AC.UK
University of Cambridge

John Bronskill[*]
University of Cambridge

James Requiema
University of Toronto, Vector Institute

Shoaib Ahmed Siddiqui
University of Cambridge

Javier González
Microsoft Research Cambridge

David Duvenaud
University of Toronto, Vector Institute

Richard E. Turner
University of Cambridge The Alan Turing Institute

Abstract

We introduce a simple method for probabilistic predictions on tabular data based on Large Language Models (LLMs) called JoLT (Joint LLM Process for Tabular data). JoLT uses the in-context learning capabilities of LLMs to define joint distributions over tabular data conditioned on user-specified side information about the problem, exploiting the vast repository of latent problem-relevant knowledge encoded in LLMs. JoLT defines joint distributions for multiple target variables with potentially heterogeneous data types without any data conversion, data preprocessing, special handling of missing data, or model training, making it accessible and efficient for practitioners. Our experiments show that JoLT outperforms competitive methods on low-shot single-target and multi-target tabular classification and regression tasks. Furthermore, we show that JoLT can automatically handle missing data and perform data imputation by leveraging textual side information.

1. Introduction

When modeling tabular data, incorporating side information and domain expertise often presents a challenge, particularly for lay users without advanced technical skills. As a result, leveraging Large Language Models (LLMs) to integrate textual information for tabular data prediction (Fang et al., 2024; Lu et al., 2024) is a natural approach. Trained on vast corpora of internet data, LLMs possess substantial implicit knowledge and offer a user-friendly interface for expressing and incorporating side information through natural language. Prediction using LLMs typically uses one of two approaches – fine-tuning (Yosinski et al., 2014)

[*] Equal contribution

or inference-time in-context learning (ICL) (Brown et al., 2020). Fine-tuning involves modifying LLM's weights to adapt it for specific tasks. This approach has its own limitations, including high resource requirements, overfitting risks with small datasets, and potential privacy concerns when using sensitive data. In contrast, ICL is a simpler, more accessible alternative, enabling predictions without gradient updates to the model. This approach reduces the burden on users by eliminating the need for training or hyperparameter tuning, making it suitable for scenarios with limited data, such as personalization (Massiceti et al., 2021; Ding et al., 2017).

Requeima et al. (2024) introduced LLM Processes (LLMPs) which used pretrained LLMs and ICL to do probabilistic regression conditioned on textual side information. LLMPs effectively combine data and metadata, such as dataset descriptions and column headings, with the LLM's latent knowledge. In this work, we build upon LLMPs and present JoLT (Joint LLMP for Tabular data). JoLT extends LLMPs beyond regression to make joint probabilistic predictions for multiple target variables, accommodating both numerical and categorical data types. Our approach eliminates the need for preprocessing, missing data imputation, model training, or hyperparameter tuning, making it highly accessible to non-experts in machine learning. Additionally, JoLT empowers users to incorporate side information or provide specific instructions in plain language or text, allowing for seamless integration of expert knowledge and contextual insights from the user and LLM into the modeling process. Our main contributions are: (i) We present JoLT, a method that extends LLMPs beyond regression to make joint probabilistic predictions for multiple target variables with heterogeneous data types on tabular datasets. (ii) We demonstrate that JoLT outperforms competitive approaches on low-shot single and multiple target tabular classification and regression tasks. (iii) We show that JoLT implicitly handles missing tabular data and performs as well or better on downstream tasks when compared to imputing data as a preprocessing step. (iv) Finally, we show that JoLT can also effectively impute missing data by leveraging textual side information about the problem. Table 1 provides a summary.

Method	Automatically handle missing data	Mixed types + strings	Can use side info	In-context learning (no training)	Joint probabilities by default	Fast at test time	Scalable to large tables
TabLLM	×	✓	✓	×	×	×	×
TabPFN	×	×	×	✓	×	✓	×
XGBoost	×	×	×	×	×	✓	✓
JoLT (Ours)	✓	✓	✓	✓	✓	×	×

Table 1: JoLT: key features and comparison with competitive methods.

2. Method

In this section, we describe how to make heterogeneous multiple target probabilistic predictions in the presence of missing data using JoLT.

Prompt Engineering. In a prediction setting, tabular datasets consist of multiple rows of examples or records. Each row consists of one or more columns of features and one or more columns of targets to be predicted based on the feature values. Training examples have both observed features and observed targets, while test examples only have observed features. Fig. 1 shows how we design a prompt that contains the entire training set and the features for a test example that is fed to the LLM in order to generate a prediction.

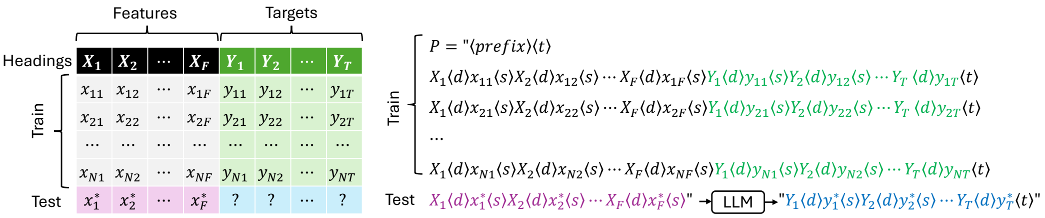
Features
Targets
Headings
X_1 X_2 ⋯ X_F Y_1 Y_2 ⋯ Y_T
Train
x_{11} x_{12} ⋯ x_{1F} y_{11} y_{12} ⋯ y_{1T}
x_{21} x_{22} ⋯ x_{2F} y_{21} y_{22} ⋯ y_{2T}
x_{N1} x_{N2} ⋯ x_{NF} y_{N1} y_{N2} ⋯ y_{NT}
Test
x_1^* x_2^* ⋯ x_F^* ? ? ⋯ ?
$P = $"$\langle prefix\rangle\langle t\rangle$
$X_1\langle d\rangle x_{11}\langle s\rangle X_2\langle d\rangle x_{12}\langle s\rangle \cdots X_F\langle d\rangle x_{1F}\langle s\rangle Y_1\langle d\rangle y_{11}\langle s\rangle Y_2\langle d\rangle y_{12}\langle s\rangle \cdots Y_T\langle d\rangle y_{1T}\langle t\rangle$
$X_1\langle d\rangle x_{21}\langle s\rangle X_2\langle d\rangle x_{22}\langle s\rangle \cdots X_F\langle d\rangle x_{2F}\langle s\rangle Y_1\langle d\rangle y_{21}\langle s\rangle Y_2\langle d\rangle y_{22}\langle s\rangle \cdots Y_T\langle d\rangle y_{2T}\langle t\rangle$
⋯
$X_1\langle d\rangle x_{N1}\langle s\rangle X_2\langle d\rangle x_{N2}\langle s\rangle \cdots X_F\langle d\rangle x_{NF}\langle s\rangle Y_1\langle d\rangle y_{N1}\langle s\rangle Y_2\langle d\rangle y_{N2}\langle s\rangle \cdots Y_T\langle d\rangle y_{NT}\langle t\rangle$
$X_1\langle d\rangle x_1^*\langle s\rangle X_2\langle d\rangle x_2^*\langle s\rangle \cdots X_F\langle d\rangle x_F^*\langle s\rangle$"
LLM
"$Y_1\langle d\rangle y_1^*\langle s\rangle Y_2\langle d\rangle y_2^*\langle s\rangle \cdots Y_T\langle d\rangle y_T^*\langle t\rangle$"

Figure 1: **Mapping tabular data to a prompt P.** The diagram on the left depicts a tabular dataset where the first N rows are training examples with F features and T targets fully observed. The last row represents a test example with F observed features and unobserved targets. The prompt at the top right is formed by serializing the N training examples and the test features into a single string. The prompt serves as input to a pretrained LLM that will generate the targets to complete the test example. See Table A.1 for nomenclature.

For each test example, we serialize each row of training data, including both features and targets, plus the features of the test example to a string. Each feature column may be of any data type (e.g. numerical - integer or floating point, categorical, date/time, addresses, unstructured text, etc.) given that the type is or can be converted to a string. The target columns can also be of any type. We do not perform any preprocessing or scaling of the data as we want to retain the natural scale and units of the data such that related knowledge encoded in the LLM can be leveraged. Importantly, we do not modify the LLM weights via training or fine-tuning.

Prediction. Here we present two approaches to making predictions on multiple target, heterogeneous data – rejection sampling and sampling from a full distribution.

Rejection Sampling. After feeding the prompt to the LLM, we use the autoregressive token prediction capability of the LLM to generate the targets for the test example. Knowing the number and types of the targets, we can parse the generated output and ensure it conforms to the expected format. If not, we reject the sample. For numerical targets, we ensure the sample contains a valid number, and for categorical targets, we ensure that it contains a known category. For a point estimate of a quantity, we can take a single top 1 sample from the LLM. To obtain a point estimate and uncertainty for numerical targets, we can take a set of samples and use the median for the point estimate and compute a confidence interval over the range. For categorical targets, the set of samples form a categorical distribution and the point estimate is the category with the most samples.

Full Distribution via LLM Logits. We can compute the probability of a target value s that is a member of the set of possible target values $\mathcal{S}$ conditioned on a prompt P as: $p(y = s|P, s \in S) = p(y = s|P)/\sum_{s' \in \mathcal{S}} p(y = s'|P)$. For categorical targets, we can obtain $p(y = s|P)$ for each category s from the LLM logits using Algorithm B.1 and then use the expression above to get the normalized probability for each category. We can then predict the target category that has the highest probability or sample from the resulting full categorical distribution. This approach is generally preferable to sampling when $|\mathcal{S}|$ is relatively small, as it grants direct access to the full predictive distribution while remaining computationally tractable.

Computing Joint Predictive Distributions. Using the product rule, we can compute the joint probability of the ground truth target values for any test example as:

$$p(y_1^*, y_2^*, \ldots, y_T^*) = p(y_1^*|\text{“}PY_1\langle d\rangle\text{”})p(y_2^*|\text{“}PY_1\langle d\rangle y_1^*\langle s\rangle Y_2\langle d\rangle\text{”})\ldots$$
$$p(y_T^*|\text{“}PY_1\langle d\rangle y_1^*\langle s\rangle Y_2\langle d\rangle y_2^*\langle s\rangle \ldots Y_T\langle d\rangle\text{”}) \quad (1)$$

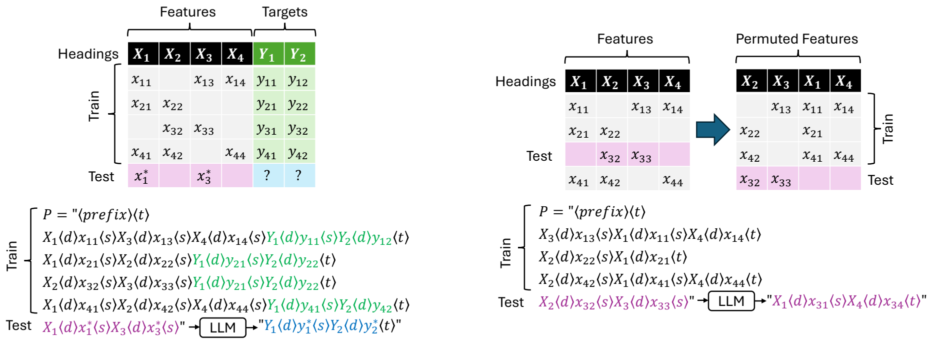
Features
Targets
Headings
X_1 X_2 X_3 X_4 Y_1 Y_2
Train
Test
x_{11} x_{13} x_{14} y_{11} y_{12}
x_{21} x_{22} y_{21} y_{22}
x_{32} x_{33} y_{31} y_{32}
x_{41} x_{42} x_{44} y_{41} y_{42}
x_1^* x_3^* ? ?
$P =$ "$\langle prefix\rangle\langle t\rangle$
$X_1\langle d\rangle x_{11}\langle s\rangle X_3\langle d\rangle x_{13}\langle s\rangle X_4\langle d\rangle x_{14}\langle s\rangle Y_1\langle d\rangle y_{11}\langle s\rangle Y_2\langle d\rangle y_{12}\langle t\rangle$
$X_1\langle d\rangle x_{21}\langle s\rangle X_2\langle d\rangle x_{22}\langle s\rangle Y_1\langle d\rangle y_{21}\langle s\rangle Y_2\langle d\rangle y_{22}\langle t\rangle$
$X_2\langle d\rangle x_{32}\langle s\rangle X_3\langle d\rangle x_{33}\langle s\rangle Y_1\langle d\rangle y_{21}\langle s\rangle Y_2\langle d\rangle y_{22}\langle t\rangle$
$X_1\langle d\rangle x_{41}\langle s\rangle X_2\langle d\rangle x_{42}\langle s\rangle X_4\langle d\rangle x_{44}\langle s\rangle Y_1\langle d\rangle y_{41}\langle s\rangle Y_2\langle d\rangle y_{42}\langle t\rangle$
$X_1\langle d\rangle x_1^*\langle s\rangle X_3\langle d\rangle x_3^*\langle s\rangle$" → LLM → "$Y_1\langle d\rangle y_1^*\langle s\rangle Y_2\langle d\rangle y_2^*\langle t\rangle$"
Features
Permuted Features
Headings
X_1 X_2 X_3 X_4
X_2 X_3 X_1 X_4
x_{11} x_{13} x_{14}
x_{21} x_{22}
x_{32} x_{33}
x_{41} x_{42} x_{44}
x_{13} x_{11} x_{14}
x_{22} x_{21}
x_{42} x_{41} x_{44}
x_{32} x_{33}
Train
Test
$P =$ "$\langle prefix\rangle\langle t\rangle$
$X_3\langle d\rangle x_{13}\langle s\rangle X_1\langle d\rangle x_{11}\langle s\rangle X_4\langle d\rangle x_{14}\langle t\rangle$
$X_2\langle d\rangle x_{22}\langle s\rangle X_1\langle d\rangle x_{21}\langle t\rangle$
$X_2\langle d\rangle x_{42}\langle s\rangle X_1\langle d\rangle x_{41}\langle s\rangle X_4\langle d\rangle x_{44}\langle t\rangle$
$X_2\langle d\rangle x_{32}\langle s\rangle X_3\langle d\rangle x_{33}\langle s\rangle$" → LLM → "$X_1\langle d\rangle x_{31}\langle s\rangle X_4\langle d\rangle x_{34}\langle t\rangle$"

Figure 2: *Left* **Missing data handling.** The diagram depicts a tabular dataset with 4 rows of training examples. The last row represents a test example with 2 unobserved targets. Empty cells represent 40% (8 out of 20 feature cells) missing completely-at-random data. The prompt is formed by simply omitting missing cells. *Right* **JoLT imputation.** To impute values for a specific row (in pink), the features are reordered such that the features with existing values for the specific row are positioned first. The prompt is constructed akin to *Left*, where instead of predicting targets Y_i, the model predicts the missing values for a specific row (e.g., columns X_1 and X_4 in the diagram).

where we have used the notation introduced in Fig. 1. Eq. (1) suggests that we can compute the joint likelihood of a multiple target test example by computing the product of the probability of each individual target conditioned on the text that precedes it using the logits of the LLM which hold token probabilities. For a numerical target, we follow Requeima et al. (2024) and approximate the probability of a ground truth value using Algorithm B.2. If the target is categorical, we can use Algorithm B.1, with y set to the true target category. In our experiments, for each test example, we compute the probability for each target, then compute the joint probability using Eq. (1), and report the negative log likelihood (NLL) as the mean of the joint NLLs over the test set.

Missing Data Handling. Our approach implicitly handles missing feature data in both training and test sets. The strategy is to simply omit any missing feature data when building the prompt as shown in Fig. 2 *Left*.

Missing Data Imputation. Our method can be extended to perform data imputation. The approach involves imputing missing values row by row, while leveraging other incomplete rows of the table as in-context training data for the LLM. To impute missing values for a specific row, the columns of the table are first permuted so that the columns containing missing values for that row are positioned after those with non-missing values. Following the procedure described above, a separate prompt is constructed for each row, and the LLM predicts the missing values conditioned on the prompt. See Fig. 2 *Right*.

3. Experiments

In this section, we evaluate the performance of JoLT prediction on single- and multiple-target low-shot tabular prediction tasks.

Classification Setting. In this experiment, we compare JoLT low-shot classification performance using the same nine datasets as Hegselmann et al. (2023b). Fig. F.1 shows that on 7 of the 9 datasets, JoLT is able to outperform, often by a large margin, boosted decision trees (XGBoost), LLM fine-tuning (TabLLM), and TabPFN (which also uses ICL, but no LLM) in the low-shot classification setting by utilizing column header side information.

Multi-target Prediction. Here, we evaluate the ability of JoLT to predict multiple heterogeneous targets and compute joint distributions on three different datasets.

Wine Quality. The first dataset is the Wine Quality dataset (Cortez et al., 2009) where there are two targets - one numerical (Alcohol %) and one categorical (Wine Quality on the scale of 0 to 10). This dataset is primarily comprised of numerical features, with the color feature column being the only categorical variable. A sample JoLT prompt is shown in Appendix D.1. We compare JoLT to two competitive methods that can make probabilistic predictions - TabPFN (Hollmann et al., 2025, 2024) and a Gaussian Process (GP). To produce multiple targets with TabPFN and the GP, we query the models autoregressively with multiple passes. We use two variants of the LLM. One that uses both prefix text $\langle prefix \rangle$ and text from the column headers X_j, Y_j, and a no text version that does not. Fig. 3 indicates that JoLT without using textual information performs poorly across the board. When using text, JoLT has the best classification accuracy, and the best NLL up to 40 shots, but loses out to TabPFN on MAE after 20 shots and to the GP at 30 shots and beyond. In all cases, TabPFN and the GP improve rapidly with increasing number of shots. When leveraging text, JoLT outperforms other models in the low-shot setting, but as the amount of training data increases, TabPFN dominates in terms of performance.

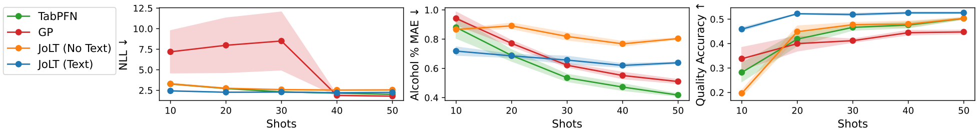

Figure 3: Results for the Wine Quality dataset as a function of shots when evaluating on 1000 test examples. The first target column is numerical (Alcohol %) using the metric MAE and the second target column is categorical (Quality on a scale of 1 to 10) using accuracy as the metric. The joint NLL is over both targets. The JoLT methods use the Gemma-2-27B LLM. The solid line and dots indicate the mean over 5 seeds which affect the training shot and test example selection and the shaded region shows a confidence interval of one σ. Tabular results are in Table F.3.

Movies. The second dataset is a subset of the Movies Box Office Dataset (2000-2024) (Jilla, 2024) where we predict the movie rating as a continuous value on a scale of 0 to 10, as well as eight binary categorical variables that indicate the movie genre, based on features that include the movie title and worldwide box office revenue. We only used the 2024 data and split the dataset (89 train, 99 test examples) so that the movies in the test set have a release date after the release date of the Gemma-2 LLM. A sample JoLT prompt is shown in Appendix D.2. The results are shown in Table 2. When predicting the numerical rating, both JoLT and TabPFN have the same error. However, JoLT outperforms TabPFN by a large margin in terms of AUC and NLL in predicting the genre attributes of the movies. This is due to JoLT being able to use the text of the movie title to help predict the binary genre targets, whereas TabPFN is not able to use text columns and hence performs poorly.

Method	Rating (MAE ↓)	Adventure (AUC ↑)	Comedy (AUC ↑)	Family (AUC ↑)	Action (AUC ↑)	Fantasy (AUC ↑)	Thriller (AUC ↑)	Drama (AUC ↑)	Horror (AUC ↑)	Mean (AUC ↑)	NLL ↓
TabPFN	**1.09**	0.55	0.59	0.55	0.64	0.57	0.32	0.55	0.52	0.54	5.59
JoLT	**1.09**	**0.87**	**0.94**	**0.97**	**0.93**	**0.94**	**0.82**	**0.85**	**0.99**	**0.91**	**4.02**

Table 2: **Movie Results**. JoLT used Gemma-2-27B. Bold indicates the highest results.

Medals. The third dataset is the Olympic Games dataset collection (Ismail, 2024) where we use the bronze, silver, and gold medal counts of 10 countries from the 1996 to 2020 summer Olympics to train on. The goal is to predict the silver and gold medal counts for ten counties at the 2024 Olympics which were held after the release date of the Gemma-2 LLM that JoLT uses. A sample prompt is in Appendix D.3. Results are shown in Table 3. JoLT has lower MAE and NLL when predicting the 2024 silver and gold medal counts as it can use the name of the country as context, whereas TabPFN only gets a numerical label.

Method	Silver (MAE ↓)	Gold (MAE ↓)	NLL ↓
TabPFN	4.16	4.91	5.89
JoLT	**3.50**	**4.80**	**1.87**

Table 3: **Medals Results**. Bold indicates the highest results.

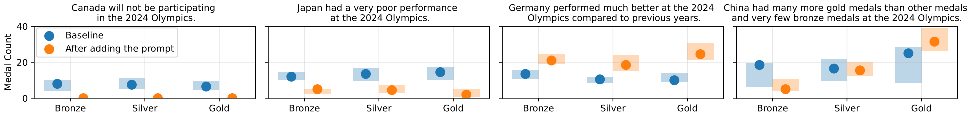

Figure 4: Effect of side information on JoLT's predictive distribution for the Medals Dataset. Each subplot corresponds to a different $\langle prefix \rangle$, shown in the title, and represents the distributions for the country mentioned in the $\langle prefix \rangle$. The baseline predictive distribution is in blue, while the distribution after adding side information is in orange. Shaded regions indicate the 25th and 75th percentiles, and dots represent the median values.

Side Information Influence. In this section, we evaluate the effect of incorporating side textual information on JoLT's predictive distribution. Using the Medals dataset, we predict the counts for bronze, silver, and gold medals. In Fig. 4, we apply four different $\langle prefix \rangle$ values that characterize a country's performance at the 2024 Olympics. In all cases, the resulting distribution shifts to better align with the textual information, demonstrating the effectiveness of textual side information in refining JoLT's predictions.

Handling Missing Data. In these experiments, we compare the *omit* strategy for handling missing data, as outlined in Section 2, with a preprocessing-based imputation method. To impute missing data, we replace missing numerical and categorical values with the column-wise mean and the column-wise mode from the training data, respectfully. We use two different datasets – the Wine Quality dataset as used in Section 3 whose features are primarily numerical, and the Car dataset whose features are all categorical.

Figs. F.2 and F.3 show the results for the two datasets where the amount of data missing completely-at-random (MCAR) varies from 10% to 40% in both the training and test features. We evaluate four JoLT variants – omit and impute, each with or without text, and TabPFN. The variants that do not use side text information perform the worst on both datasets. The results show that JoLT implicitly handles missing data, obviating the need to impute as a preprocessing step, which greatly simplifies the data science workflow.

Missing Data Imputation. While JoLT gracefully handles missing data while predicting, it is often required to recover missing feature values. In this experiment, we demonstrate JoLT's ability to use contextual information and predict multiple values seamlessly to improve imputation performance. We use the Paris 2024 data from the Olympic Games dataset collection (Ismail, 2024) which lists medals won by each of the 91 nations participating in the event. We use the following columns: Country, Gold, Silver, and Bronze. We then eliminate a fixed percentage of the data completely-at-random. The country column cannot be used by imputation methods that only use numerical data, but can be leveraged by JoLT.

An example prompt is shown in Appendix D.4. The results are shown in Fig. 5 where JoLT (gemma-2-27B) has significantly lower error than conventional imputation techniques that include MICE (Little and Rubin, 2002), k-Nearest Neighbors (Troyanskaya et al., 2001), mean, and iterative application of a Bayesian Ridge regressor.

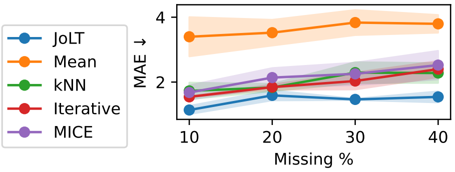

Figure 5: **JoLT imputation.** MAE as a function of % of missing data (MCAR) on the Medals dataset for JoLT and four competitive methods. The dots are the mean over 3 seeds which affect the missing pattern. The shaded region is a σ-confidence interval. Tabular results are in Table F.6.

Acknowledgements

James Requeima and David Duvenaud acknowledge funding from the Data Sciences Institute at the University of Toronto and the Vector Institute. Aliaksandra Shysheya, John Bronskill, and Richard E. Turner are supported by an EPSRC Prosperity Partnership EP/T005386/1 between the EPSRC, Microsoft Research and the University of Cambridge. Richard E. Turner is also supported by Google, Amazon, ARM, Improbable, and the EPSRC Probabilistic AI Hub (ProbAI, EP/Y028783/1).

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

Anonymous. Context-driven missing data imputation via large language model. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=b2oLgk5XRE. under review.

Barry Becker and Ronny Kohavi. Adult census income. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Marko Bohanec. Car Evaluation. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C5JP48.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

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.

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*, pages 785–794, 2016.

Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5): 304–310, 1989.

Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3571–3580, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/253614bbac999b38b5b60cae531c4969-Abstract.html.

Zhicheng Ding, Jiahao Tian, Zhenkai Wang, Jinman Zhao, and Siyang Li. Data imputation using large language model to accelerate recommendation system. 2024. URL https://api.semanticscholar.org/CorpusID:271213203.

Xi Fang, Weijie Xu, Fiona Anting Tan, Ziqing Hu, Jiani Zhang, Yanjun Qi, Srinivasan H. Sengamedu, and Christos Faloutsos. Large language models (LLMs) on tabular data: Prediction, generation, and understanding - a survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=IZnrCGF9WI.

Ahatsham Hayat and Mohammad Rashedul Hasan. Claim your data: Enhancing imputation accuracy with contextual large language models, 2024. URL https://arxiv.org/abs/2405.17712.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm. https://github.com/clinicalml/TabLLM, 2023a.

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*, pages 5549–5581. PMLR, 2023b.

Hans Hofmann. Statlog (german credit data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Tabpfn. https://github.com/PriorLabs/TabPFN, 2024.

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*, 01 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

Youssef Ismail. Olympic games (1994-2024), 2024. Retrieved January 19, 2025 from https://www.kaggle.com/datasets/youssefismail20/olympic-games-1994-2024.

Bahram Jafrasteh, Daniel Hernández-Lobato, Simón Pedro Lubián-López, and Isabel Benavente-Fernández. Gaussian processes for missing value imputation. *Know.-Based Syst.*, 273(C), August 2023. ISSN 0950-7051. doi: 10.1016/j.knosys.2023.110603. URL https://doi.org/10.1016/j.knosys.2023.110603.

Aditya Jilla. Movies box office dataset (2000-2024), 2024. Retrieved January 19, 2025 from https://www.kaggle.com/datasets/aditya126/movies-box-office-dataset-2000-2024.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. Carte: pretraining and transfer for tabular learning. In *Forty-first International Conference on Machine Learning*, 2024.

R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 2002. ISBN 9780471183860. URL http://books.google.com/books?id=aYPwAAAAMAAJ.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. Large language model for table processing: A survey. *arXiv preprint arXiv:2402.05121*, 2024.

Daniela Massiceti, Luisa M. Zintgraf, John Bronskill, Lida Theodorou, Matthew Tobias Harris, Edward Cutrell, Cecily Morrison, Katja Hofmann, and Simone Stumpf. ORBIT: A real-world few-shot dataset for teachable object recognition. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 10798–10808. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01064. URL https://doi.org/10.1109/ICCV48922.2021.01064.

Pablo Moreno-Muñoz, Antonio Artés, and Mauricio Alvarez. Heterogeneous multi-output gaussian process prediction. *Advances in neural information processing systems*, 31, 2018.

S. Moro, P. Rita, and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K306.

Roger B Nelsen. *An introduction to copulas.* Lecture notes in statistics (Springer-Verlag) ; v. 139. Springer, New York, 1999. ISBN 0387986235.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. Technical report, Louisiana State University, 1997.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

James Requeima, John Bronskill, Dami Choi, Richard E. Turner, and David Duvenaud. Llm processes: Numerical predictive distributions conditioned on natural language, 2024.

Jack W Smith, James E Everhart, William C Dickson, William C Knowler, and Robert S Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265, 1988.

Daniel J. Stekhoven and Peter Bühlmann. Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28 1:112–8, 2011. URL https://api.semanticscholar.org/CorpusID:2089531.

Olga G. Troyanskaya, Michael N. Cantor, Gavin Sherlock, Patrick O. Brown, Trevor J. Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17 6:520–5, 2001. URL https://api.semanticscholar.org/CorpusID:1105917.

Jan van Rijn and Jonathan Vis. Endgame analysis of dou shou qi. *ICGA journal*, 37: 120–124, 06 2014. doi: 10.3233/ICG-2014-37208.

I-Cheng Yeh. Blood Transfusion Service Center. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5GS39.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Yuxuan Zhao and Madeleine Udell. Matrix completion with quantified uncertainty through low rank gaussian copula. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Appendix A. Nomenclature

Table A.1: Nomenclature used in prompt construction.

Symbol	Description
P	prompt
F	number of features
T	number of targets
N	number of training examples or shots
M	number of text examples
$X_1, X_2, \ldots, X_F$	text heading for feature columns
$Y_1, Y_2, \ldots, Y_T$	text heading for target columns
$x_{i,j}$	jth feature value of the ith training example
$y_{i,j}$	jth target value of the ith training example
$x^*_{i,j}$	jth feature value of the ith test example
$y^*_{i,j}$	jth target value of the ith test example
$\langle prefix \rangle$	text string with side information
$\langle d \rangle$	separates X_j and $x_{i,j}$ or Y_j and $y_{i,j}$
$\langle s \rangle$	separates $X_j \langle d \rangle x_{i,j}$ and $X_{j+1} \langle d \rangle x_{i,j+1}$ or $Y_j \langle d \rangle y_{i,j}$ and $Y_{j+1} \langle d \rangle y_{i,j+1}$
$\langle t \rangle$	separates examples

Appendix B. Algorithms

Algorithm B.1 Computing the log probability distribution function of a categorical target y

```
Input: $\mathcal{M}$: LLM model with vocabulary $\mathcal{V}$
Input: $\mathcal{T}$: tokenizer
Input: $\mathcal{S}$: text string preceding $y$ to condition on
Input: $\mathcal{Y} = \{y_1, \ldots, y_L\}$: set of possible classes
$\mathcal{S}^T \leftarrow \mathcal{T}(\mathcal{S})$                                   // Tokenize $\mathcal{S}$
for $i \leftarrow 1$ to $L$ do
    $y_i^T \leftarrow \mathcal{T}(y_i)$                                  // Tokenize $y_i$
    $l_i \leftarrow |y_i^T|$                                     // Number of tokens in $y_i^T$
    logits $\leftarrow \mathcal{M}(\mathcal{S}^T + y_i^T)$      // Forward pass; shape of logits = $|\mathcal{S}^T + y_i^T| \times |\mathcal{V}|$
    for $j \leftarrow 1$ to $l_i$ do
        y_logits[j] $\leftarrow$ logits[$-(l_i + 2) + j,\ y_i^T[j]$]        // Logits for token $j$ in $y_i^T$
    end
    class_logits[i] $\leftarrow$ y_logits.sum                 // Logits corresponding to class $y_i$
end
y_log_pdf $\leftarrow$ CrossEntropy(logits=class_logits, target=$y$)
```

Algorithm B.2 Computing the log probability distribution function of a numerical target y (Requeima et al., 2024)

```
Input: $\mathcal{M}$: LLM model; $\mathcal{T}$: tokenizer
Input: $n$: number of digits after the decimal point for $y$
Input: $\mathcal{A} = \{$“0”, “1”, . . . , “9”, “ − ”, “.”, “$\langle t \rangle$”, “$\langle s \rangle$”$\}$: allowed characters for $y$
Input: $\mathcal{S}$: text string preceding $y$ to condition on
non_numeric_mask ← all_tokens $\notin \mathcal{T}(\mathcal{A})$    // Mask of non-numeric tokens
$y^T \leftarrow \mathcal{T}(\texttt{str}(y))$    // Tokenize $y$
$l \leftarrow |y^T|$    // Number of tokens in $y^T$
logits ← $\mathcal{M}(\mathcal{T}(\mathcal{S}) + y^T)$    // Run the LLM model forward
y_logits ← logits$[-l-1 : -1]$    // Logits corresponding to $y$
y_logits[non_numeric_mask] ← -100    // Mask out non-numeric tokens
y_log_pmf ← CrossEntropy(logits=y_logits, targets=$y^T$).sum    // Probability mass of
  bin that includes $y$
y_log_pdf ← y_log_pmf + $n \log 10$    // Convert probability mass to continuous
  likelihood
```

Appendix C. Related Work

In this section, we survey related methods for tabular prediction and imputation. For a more complete treatment, refer to Borisov et al. (2022) for a survey of deep learning methods for tabular data and Fang et al. (2024); Lu et al. (2024) for applying LLMs to tabular data.

Classification and Multi-target prediction TabLLM (Hegselmann et al., 2023b) fine-tunes an LLM for single target classification only. It can incorporate side information, but performance is limited in the few-shot setting. TabPFN (Hollmann et al., 2025) is an effective ICL method for both classification and regression and offers uncertainty in the form of quantiles. However, it is restricted to handling numerical and categorical data, is unable to incorporate text or side information, cannot predict multiple targets at once, and does not perform well in the low-shot setting. LLM Processes (Requeima et al., 2024) support multi-target regression with uncertainty estimates and side information but do not handle classification or heterogeneous targets. JoLT extends LLM Processes by enabling classification and supporting heterogeneous multiple targets in the tabular data setting. Carte (Kim et al., 2024) uses a graph-attentional network pretrained across multiple tables, and subsequently fine-tuned on a downstream dataset. It can incorporate side information while performing single-target regression or classification, but does not provide uncertainty. Finally, GPs can make multiple target probabilistic predictions on heterogeneous data (Moreno-Muñoz et al., 2018), but require training and cannot easily incorporate side information.

Handling Missing Data and Data Imputation There exists a myriad of widely used imputation methods, including mean, median, and mode imputation, k-Nearest Neighbors (Troyanskaya et al., 2001), MICE (Little and Rubin, 2002), and tree-based methods like MissForest (Stekhoven and Bühlmann, 2011). However, they are restricted to using numerical or categorical data, and hence, are unable to exploit side information. Furthermore, they do not provide any estimates of uncertainty or distributions for the imputed values. In contrast, Bayesian methods, such as Gaussian Processes (GPs) (Rasmussen and Williams,

2006) and Gaussian Copula (Nelsen, 1999), offer the advantage of providing uncertainty estimates. While Bayesian methods have been extended to handle missing data (Zhao and Udell, 2020; Jafrasteh et al., 2023), their widespread adoption is hindered by scalability limitations and the challenges associated with incorporating side information effectively. Recent work has also explored the use of LLMs for data imputation. Anonymous (2024) proposes a nearest-neighbor-based imputation method that operates in the embedding space of an LLM. However, this method does not provide uncertainty estimates for the imputed values. Ding et al. (2024) and Hayat and Hasan (2024) fine-tune LLMs to handle missing data and report notable improvements on several downstream tasks. To the best of our knowledge, there are no methods that utilize LLMs to perform data imputation in low-shot scenarios. In contrast, JoLT handles missing data automatically, and if imputed values are required, JoLT can impute missing data with uncertainty, has the ability to incorporate side information, and operate well in the low-shot setting.

Appendix D. Sample Prompts

In this section, we provide sample prompts for various experiments.

D.1. Wine Quality Experiment Sample Prompt

A sample prompt with one training example, plus test example features, and $\langle d \rangle$ = ":", $\langle s \rangle$ = ";", and $\langle t \rangle$ = "\n" is:

> "The data contains features that determine the quality of wine. Predict the alcohol content and the quality score of each wine based on the features.\nfixed_acidity: 6.2; volatile_acidity: 0.23; citric_acid: 0.35; residual_sugar: 0.7; chlorides: 0.051; free_sulfur_dioxide: 24.0; total_sulfur_dioxide: 111.0; density: 0.992; pH: 3.37; sulphates: 0.43; color: white; alcohol: 11.0; quality: 3\nfixed_acidity: 9.9; volatile_acidity: 0.49; citric_acid: 0.23; residual_sugar: 2.4; chlorides: 0.087; free_sulfur_dioxide: 19.0; total_sulfur_dioxide: 115.0; density: 0.995; pH: 2.77; sulphates: 0.44; color: white;"

D.2. Movies Box Office Sample Prompt

A sample prompt with one training example, plus test example features, and $\langle d \rangle$ = ":", $\langle s \rangle$ = ";", and $\langle t \rangle$ = "\n" is:

> "Each example contains 10 columns: Movie Name, Revenue in Millions of Dollars, Rating, and 8 genre tags (Adventure, Comedy, Family, Action, Fantasy, Thriller, Drama, and Horror). Predict the movie rating and genre tags.\nMovie Name:Kung Fu Panda 4;ID:36;Revenue in $Millions:547.7;Rating:7.1;Adventure:No;Comedy:No;Family:Yes;Action:Yes;Fantasy:Yes; Thriller:No;Drama:No;Horror:No\nMovie Name:Speak No Evil;ID:120;Revenue in $Millions:76.8;"

D.3. Olympic Games Dataset Prompt

A sample prompt with one training example, plus test example features, and $\langle d \rangle$ = “:”, $\langle s \rangle$ = “;”, and $\langle t \rangle$ = “\n” is:

> “Each example contains five columns: Olympic Year, Country, Bronze Medal Count, Silver Medal Count, and Gold Medal Count that describe what type and how many medals a country won at the Olympic games that year. Predict the number of silver and gold medals won by that country in that year.\nOlympic Year:2020;Country:Netherlands;Bronze Medal Count:14;Silver Medal Count:12;Gold Medal Count:10\nOlympic Year:2024;Country:USA;Bronze Medal Count:42;’”

D.4. Imputation Prompt

A sample prompt with one training example, plus test example features, and $\langle d \rangle$ = “:”, $\langle s \rangle$ = “;”, and $\langle t \rangle$ = “\n” is:

> “Each example contains four columns: Country, Silver Medal Count, Bronze Medal Count, and Gold Medal Count that describe what type and how many medals a country won at the Paris 2024 olympics.\nCountry:Thailand;Silver Medal Count:3;Bronze Medal Count:2;Gold Medal Count:1\nCountry:Slovakia;Silver Medal Count:0;Bronze Medal Count:1;”

Appendix E. Datasets

Classification For the few-shot classification experiments, we utilize the nine datasets introduced in Hegselmann et al. (2023b). The serialized versions of these datasets are obtained from Hegselmann et al. (2023a). These datasets are:

- **Bank** (Moro et al., 2014) contains records relevant to a direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (binary classification). It consists of $45,211$ rows and 16 features; $5,289$ labels are positive.

- **Blood** (Yeh, 2008) contains 748 donor records from the Blood Transfusion Service Center in Taiwan. The classification task is to predict whether a donor returned for another blood donation.

- **Calhousing** (Pace and Barry, 1997) has entries of houses found in a given California district and some summary stats about them based on the 1990 U.S. census data. Following (Hegselmann et al., 2023a), the goal is to predict whether the house value is below or above the median in its district.

- **Cars** (Bohanec, 1988) contains records of various cars, characterized by six attributes. The task is a multiclass classification problem to evaluate the state of each car.

- **Creditg** (Hofmann, 1994) contains 1000 records, each described by 20 attributes, and the task is to classify individuals as either good or bad credit risks. Of these records, 700 are classified as good credit risks.

- **Diabetes** (Smith et al., 1988): originally from the National Institute of Diabetes and Digestive and Kidney Diseases, this dataset aims to diagnostically predict whether a patient has diabetes based on specific diagnostic measurements. Among the records, 268 cases are positive for diabetes.

- **Heart** (Detrano et al., 1989) combines records from four hospitals in Cleveland, Hungary, Switzerland, and Long Beach V. The task is to predict the presence of heart disease in patients. Among the 918 patients, 508 are diagnosed as positive for heart disease.

- **Income** (Becker and Kohavi, 1996) contains records of $48,842$ individuals with 12 attributes collected in the 1994 U.S. Census. The classification task is to predict whether annual income of an individual exceeds $50K. The dataset has $11,687$ positive labels.

- **Jungle** (van Rijn and Vis, 2014) consists of $44,819$ endgame positions from Jungle Chess. Each position is described by 6 attributes, and the task is to predict whether the white player wins. Among the records, $23,062$ are positive outcomes for the white player.

Multi-target prediction For the multi-target prediction in Section 3, we use the following datasets:

- **Wine Quality** (Cortez et al., 2009) contains records of Portuguese wines, each characterized by 11 physiochemical attributes. The original task is to predict wine quality on a scale from 0 to 10. To evaluate multi-target prediction capabilities, we modified the task to predict both the wine quality (categorical) and alcohol percentage (numerical).

- **Movies Box office Dataset (2000-2024)** (Jilla, 2024) provides a comprehensive analysis of global box office performance from 2000 to 2024. Each row represents a movie and includes attributes such as release year, genres, production budget, worldwide gross, and additional descriptive features. The dataset contains 4955 movies.

- **Olympic Games (1994-2024)** (Ismail, 2024) is a database of medals awarded at the Summer and Winter Olympic Games from 1994 to 2024. Each table in the database corresponds to a specific Olympic Games, detailing medal counts by participating countries. Each row includes the country code and the number of gold, silver, and bronze medals won by the respective country.

In Section 3, we evaluate using the **Cars** and **Wine Quality** datasets, while in Section 3, we use a subset of the **Olympic Games** dataset.

Appendix F. Additional Experimental Results

In this section, we expand on the main paper by providing additional experimental results, along with tabular versions of the findings originally presented as plots. For any experiment with a regression target, we use *top*-1 sampling. The one exception is in the Movies experiment, where we use 50 samples and take the median as the point estimate.

F.1. Classification

In this section, we compare JoLT classification performance using datasets from Hegselmann et al. (2023b). To make a fair comparison, we use the same serialized datasets provided by Hegselmann et al. (2023a). The datasets used in this experiment are described in Appendix E. We consider XGBoost (Chen and Guestrin, 2016), TabPFN (Hollmann et al., 2025), and TabLLM (Hegselmann et al., 2023b) as competitive baselines for comparison. Like JoLT, TabLLM also relies on an LLM for prediction. However, TabLLM utilizes fine-tuning on the training data as opposed to ICL, which is the main focus of JoLT. Fig. F.1 shows that on 7 of the 9 datasets, JoLT is able to outperform, often by a large margin, boosted decision trees (XGBoost), LLM fine-tuning (TabLLM), and TabPFN (which also uses ICL, but no LLM) in the low-shot classification setting by utilizing column header side information. However, expectedly, the gap shrinks with an increasing number of shots. Table F.2 presents the tabular version of Fig. F.1.

Summary: JoLT is able to outperform boosted decision trees (XGBoost), LLM fine-tuning (TabLLM), and TabPFN (which also uses ICL, but without an LLM) in the low-shot classification setting by utilizing column header side information. XGBoost and TabPFN cannot use side information and rely on larger amounts of training data to perform well. TabLLM can leverage text information, but tends to overfit when fine-tuning on a small amount of training data.

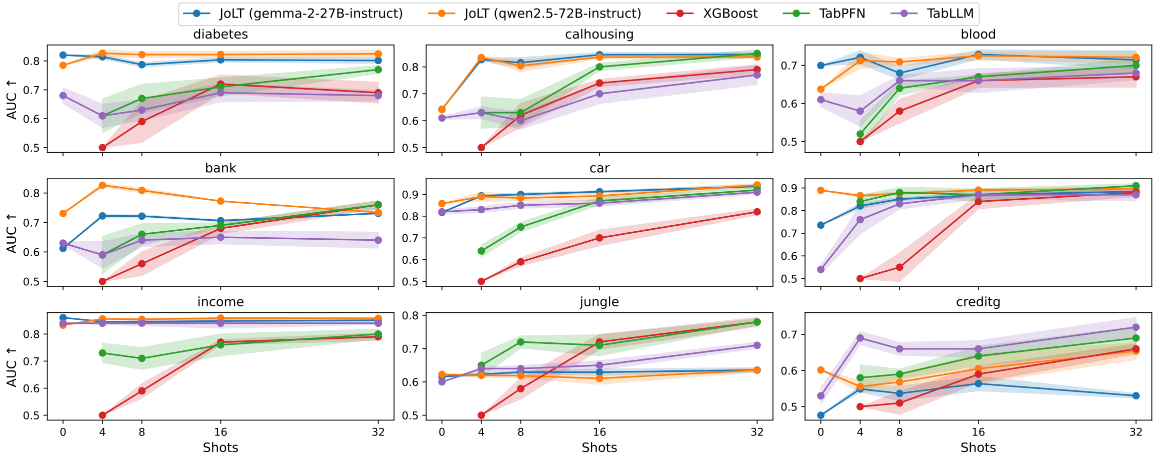

Figure F.1: Area Under the Receiver Operating Characteristic Curve (AUC) as a function of shot for JoLT using two different LLMs and three competitive methods. The solid line and dots indicate the mean over 5 seeds which affect the training shot selection and the shaded region shows a confidence interval of one σ. Competitive data from Hegselmann et al. (2023b). Tabular results are in Table F.2.

Table F.2: **Classification**. AUC as a function of shot for three JoLT configurations and three competitive methods. Values are the mean and 95% confidence interval (CI) over 5 seeds that affect the training shot selection. Due to limited computational resources values at 16 and 32 shots with 0 CI use only a single seed. Competitive data from (Hegselmann et al., 2023b).

		Shot				
Dataset	**Method**	**0**	**4**	**8**	**16**	**32**
bank	XGBoost	-	0.5±0.00	0.56±0.08	0.68±0.04	0.76±0.03
	TabPFN	-	0.59±0.12	0.66±0.07	0.69±0.02	0.76±0.03
	TabLLM	0.63±0.01	0.59±0.09	0.64±0.04	0.65±0.04	0.64±0.05
	JoLT (Gemma-2-2B)	0.46±0.00	0.62±0.05	0.62±0.05	0.57±0.05	0.52±0.09
	JoLT (Gemma-2-27B)	0.61±0.00	0.72±0.01	0.72±0.01	0.71±0.01	0.73±0.00
	JoLT (Qwen-2.5-72B)	0.73±0.00	0.83±0.01	0.81±0.02	0.77±0.00	0.73±0.00
blood	XGBoost	-	0.5±0.00	0.58±0.06	0.66±0.04	0.67±0.05
	TabPFN	-	0.52±0.07	0.64±0.04	0.67±0.01	0.7±0.04
	TabLLM	0.61±0.04	0.58±0.08	0.66±0.03	0.66±0.06	0.68±0.04
	JoLT (Gemma-2-2B)	0.56±0.00	0.62±0.08	0.58±0.04	0.64±0.06	0.59±0.05
	JoLT (Gemma-2-27B)	0.70±0.00	0.72±0.03	0.68±0.03	0.73±0.02	0.71±0.05
	JoLT (Qwen-2.5-72B)	0.64±0.00	0.71±0.04	0.71±0.02	0.73±0.02	0.72±0.03
calhousing	XGBoost	-	0.5±0.00	0.62±0.09	0.74±0.03	0.79±0.04
	TabPFN	-	0.63±0.11	0.63±0.10	0.8±0.03	0.85±0.03
	TabLLM	0.61±0.01	0.63±0.04	0.6±0.06	0.7±0.07	0.77±0.07
	JoLT (Gemma-2-2B)	0.45±0.00	0.72±0.01	0.66±0.09	0.76±0.04	0.78±0.04
	JoLT (Gemma-2-27B)	0.64±0.00	0.83±0.01	0.82±0.04	0.85±0.02	0.85±0.01
	JoLT (Qwen-2.5-72B)	0.64±0.00	0.83±0.01	0.80±0.04	0.84±0.01	0.84±0.01
car	XGBoost	-	0.5±0.00	0.59±0.04	0.7±0.07	0.82±0.03
	TabPFN	-	0.64±0.05	0.75±0.04	0.87±0.04	0.92±0.02
	TabLLM	0.82±0.02	0.83±0.03	0.85±0.03	0.86±0.03	0.91±0.02
	JoLT (Gemma-2-2B)	0.73±0.00	0.84±0.02	0.79±0.02	0.79±0.04	0.74±0.04
	JoLT (Gemma-2-27B)	0.82±0.00	0.89±0.01	0.90±0.01	0.91±0.01	0.94±0.01
	JoLT (Qwen-2.5-72B)	0.86±0.00	0.89±0.04	0.88±0.02	0.89±0.04	0.94±0.01
creditg	XGBoost	-	0.5±0.00	0.51±0.06	0.59±0.04	0.66±0.03
	TabPFN	-	0.58±0.07	0.59±0.03	0.64±0.05	0.69±0.06
	TabLLM	0.53±0.04	0.69±0.04	0.66±0.04	0.66±0.04	0.72±0.05
	JoLT (Gemma-2-2B)	0.52±0.00	0.52±0.03	0.53±0.06	0.55±0.04	0.50±0.04
	JoLT (Gemma-2-27B)	0.48±0.00	0.55±0.02	0.54±0.04	0.56±0.04	0.53±0.01
	JoLT (Qwen-2.5-72B)	0.60±0.00	0.56±0.03	0.57±0.05	0.60±0.08	0.65±0.05
diabetes	XGBoost	-	0.5±0.00	0.59±0.14	0.72±0.06	0.69±0.07
	TabPFN	-	0.61±0.11	0.67±0.10	0.71±0.06	0.77±0.03
	TabLLM	0.68±0.05	0.61±0.08	0.63±0.07	0.69±0.06	0.68±0.04
	JoLT (Gemma-2-2B)	0.62±0.00	0.73±0.06	0.71±0.06	0.76±0.02	0.73±0.06
	JoLT (Gemma-2-27B)	0.82±0.00	0.81±0.02	0.79±0.01	0.80±0.02	0.80±0.02
	JoLT (Qwen-2.5-72B)	0.78±0.00	0.83±0.02	0.82±0.02	0.82±0.02	0.82±0.02
heart	XGBoost	-	0.5±0.00	0.55±0.12	0.84±0.06	0.88±0.04
	TabPFN	-	0.84±0.05	0.88±0.04	0.87±0.05	0.91±0.02
	TabLLM	0.54±0.04	0.76±0.12	0.83±0.04	0.87±0.04	0.87±0.05
	JoLT (Gemma-2-2B)	0.64±0.00	0.74±0.01	0.80±0.04	0.72±0.08	0.65±0.09
	JoLT (Gemma-2-27B)	0.74±0.00	0.82±0.02	0.85±0.01	0.87±0.01	0.88±0.01
	JoLT (Qwen-2.5-72B)	0.89±0.00	0.87±0.01	0.88±0.01	0.89±0.01	0.90±0.02
income	XGBoost	-	0.5±0.00	0.59±0.05	0.77±0.02	0.79±0.03
	TabPFN	-	0.73±0.07	0.71±0.08	0.76±0.08	0.8±0.04
	TabLLM	0.84±0.00	0.84±0.01	0.84±0.02	0.84±0.04	0.84±0.01
	JoLT (Gemma-2-2B)	0.82±0.00	0.82±0.02	0.82±0.01	0.83±0.02	0.83±0.01
	JoLT (Gemma-2-27B)	0.86±0.00	0.85±0.00	0.85±0.01	0.85±0.01	0.85±0.00
	JoLT (Qwen-2.5-72B)	0.83±0.00	0.86±0.00	0.85±0.01	0.86±0.00	0.86±0.00
jungle	XGBoost	-	0.5±0.00	0.58±0.06	0.72±0.04	0.78±0.03
	TabPFN	-	0.65±0.07	0.72±0.04	0.71±0.06	0.78±0.02
	TabLLM	0.6±0.00	0.64±0.01	0.64±0.02	0.65±0.03	0.71±0.02
	JoLT (Gemma-2-2B)	0.67±0.00	0.60±0.02	0.59±0.04	0.55±0.04	0.61±0.04
	JoLT (Gemma-2-27B)	0.62±0.00	0.62±0.01	0.63±0.01	0.63±0.02	0.64±0.01
	JoLT (Qwen-2.5-72B)	0.62±0.00	0.62±0.01	0.62±0.01	0.61±0.03	0.64±0.01

F.2. Multi-target Prediction

This section extends Section 3 by presenting a tabular version (Table F.3) of Fig. 3, which displays the results of predicting two target columns from the Wine Quality dataset (Cortez et al., 2009) as a function of shots.

Table F.3: **Multi-target Prediction**. Results for predicting two target columns from the Wine Quality dataset (Cortez et al., 2009) as a function of shots. 1000 test examples were used. The first target column is numerical (Alcohol %) using the metric Mean Absolute Error (MAE) and the second target column is categorical (Quality on a scale of 1 to 10) using classification accuracy as the metric. The joint NLL is over both targets. The LLMP methods use the Gemma-2-27B LLM. LLMP (Text) utilized both prefix text $\langle prefix \rangle$ and text from the column headers X_j, Y_j, whereas LLMP (No Text) did not. Values are the mean and 95% confidence interval (CI) over 5 seeds that affect the training shot and test example selection.

		Shots				
Method	**Metric**	**10**	**20**	**30**	**40**	**50**
TabPFN	MAE↓	0.880±0.144	0.688±0.077	0.535±0.050	0.472±0.047	0.418±0.017
	ACC↑	0.282±0.074	0.419±0.054	0.466±0.023	0.476±0.023	0.503±0.007
	NLL↓	3.281±0.149	2.724±0.135	2.314±0.060	2.159±0.077	2.013±0.030
GP	MAE↓	0.942±0.044	0.771±0.020	0.622±0.014	0.551±0.023	0.510±0.014
	ACC↑	0.338±0.046	0.400±0.030	0.412±0.007	0.445±0.010	0.448±0.007
	NLL↓	7.185±2.573	7.980±3.322	8.499±3.532	1.879±0.068	1.795±0.021
LLMP (No text)	MAE↓	0.866±0.005	0.891±0.038	0.818±0.049	0.768±0.033	0.804±0.011
	ACC↑	0.197±0.024	0.449±0.046	0.477±0.018	0.480±0.024	0.503±0.006
	NLL↓	3.285±0.143	2.747±0.086	2.589±0.048	2.534±0.032	2.552±0.029
LLMP (Text)	MAE↓	0.718±0.056	0.686±0.022	0.657±0.053	0.620±0.025	0.638±0.011
	ACC↑	0.459±0.013	0.522±0.002	0.519±0.010	0.526±0.007	0.526±0.006
	NLL↓	2.434±0.101	2.268±0.048	2.293±0.084	2.205±0.057	2.248±0.019

F.3. Handling Missing Data

This section expands on Section 3 by reporting our findings for the Wine Quality and Car datasets. Fig. F.2 and Table F.4 show results for the Wine Quality dataset, while Fig. F.3 and Table F.5 present results for the Car dataset. In both cases, the amount of data missing completely-at-random (MCAR) varies from 10% to 40% in both the training and test features. As discussed in the main text, we evaluate four JoLT variants—omit and impute, each with or without side text information—and compare them against TabPFN. Remarkably, for the JoLT variants that use side text information, the omit approach performs almost identically to imputing data on the Wine Quality dataset and exceeds it on the Car dataset. Interestingly, the opposite is true when no side text is used — impute performs better than omit. This supports our argument that the heading text labels on each feature cell helps the LLM to do a better job of knowing what features are missing (or present) and compensate appropriately. Compared to TabPFN, JoLT omit outperforms at low shot and performs similarly at higher shot. The exception is at 40% missing using the car dataset where JoLT omit outperforms TabPFN by a large margin.

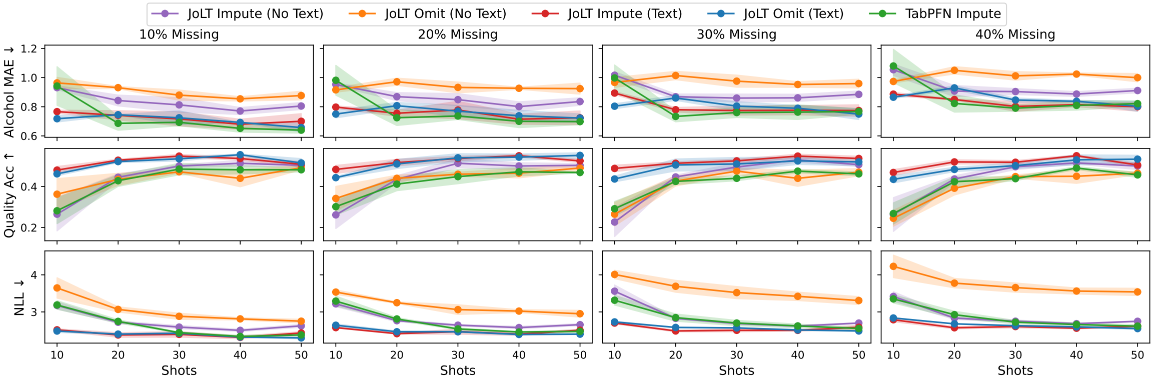

Figure F.2: Performance metrics for JoLT that uses the gemma-2-27B LLM and TabPFN as a function of shots and percentage of data missing completely-at-random (MCAR) on the multitarget Wine Quality dataset (Cortez et al., 2009). 200 test examples were used. The solid line and dots indicate the mean over 3 seeds which affect the training shot and test example selection as well as the missing pattern and the shaded region shows a confidence interval of one σ. The first target column is numerical (Alcohol %) using the metric Mean Absolute Error (MAE) and the second target column is categorical (Wine Quality on a scale of 1 to 10) using classification accuracy as the metric (ACC). The joint negative log-likelihood (NLL) is over both targets. Tabular version is in Table F.4.

Table F.4: **Missing Data Handling on Wine Quality**. Performance metrics for JoLT that uses the gemma-2-27B LLM and TabPFN as a function of shots and percentage of data missing completely-at-random (MCAR) on the Wine Quality dataset (Cortez et al., 2009). 200 test examples were used. Values are the mean and 95% confidence interval over 5 seeds that affect the training and test shot selection and the missing pattern. The first target column is numerical (Alcohol %) using the metric Mean Absolute Error (MAE) and the second target column is categorical (Wine Quality on a scale of 1 to 10) using classification accuracy as the metric (ACC). The joint negative log-likelihood (NLL) is over both targets.

			Shots				
Missing %	**Method**	**Metric**	**10**	**20**	**30**	**40**	**50**
10	LLMP Impute (No Text)	MAE	0.943±0.053	0.875±0.030	0.832±0.081	0.809±0.026	0.824±0.035
		ACC	0.309±0.120	0.475±0.048	0.480±0.014	0.470±0.051	0.473±0.023
		NLL	3.420±0.428	2.746±0.025	2.595±0.100	2.608±0.036	2.570±0.078
	LLMP Omit (No Text)	MAE	0.955±0.051	0.931±0.015	0.880±0.069	0.854±0.029	0.877±0.059
		ACC	0.370±0.100	0.435±0.058	0.472±0.012	0.440±0.081	0.492±0.019
		NLL	3.716±0.377	3.067±0.139	2.882±0.166	2.812±0.058	2.750±0.100
	LLMP Impute (Text)	MAE	0.718±0.038	0.769±0.027	0.760±0.106	0.707±0.022	0.690±0.059
		ACC	0.458±0.030	0.527±0.023	0.532±0.013	0.532±0.013	0.530±0.020
		NLL	2.458±0.058	2.409±0.043	2.428±0.093	2.347±0.020	2.356±0.041
	LLMP Omit (Text)	MAE	0.718±0.045	0.745±0.005	0.724±0.093	0.692±0.001	0.657±0.057
		ACC	0.462±0.019	0.522±0.011	0.535±0.021	0.555±0.007	0.515±0.042
		NLL	2.484±0.077	2.401±0.018	2.424±0.065	2.322±0.010	2.299±0.053
	TabPFN	MAE	0.942±0.260	0.686±0.094	0.693±0.047	0.652±0.052	0.639±0.040
		ACC	0.282±0.123	0.428±0.055	0.485±0.055	0.482±0.039	0.482±0.021
		NLL	3.188±0.227	2.745±0.090	2.441±0.028	2.339±0.060	2.379±0.056
20	LLMP Impute (No Text)	MAE	0.956±0.081	0.945±0.025	0.877±0.071	0.867±0.040	0.858±0.064
		ACC	0.284±0.091	0.467±0.049	0.487±0.019	0.473±0.072	0.495±0.012
		NLL	3.558±0.508	2.852±0.052	2.643±0.087	2.654±0.028	2.607±0.073
	LLMP Omit (No Text)	MAE	0.936±0.041	0.972±0.048	0.933±0.073	0.927±0.014	0.924±0.075
		ACC	0.351±0.088	0.440±0.030	0.460±0.020	0.465±0.045	0.490±0.030
		NLL	3.714±0.318	3.248±0.044	3.062±0.258	3.022±0.074	2.951±0.181
	LLMP Impute (Text)	MAE	0.781±0.066	0.828±0.059	0.769±0.098	0.743±0.046	0.766±0.055
		ACC	0.455±0.023	0.528±0.042	0.520±0.035	0.528±0.019	0.540±0.005
		NLL	2.574±0.086	2.492±0.053	2.474±0.107	2.409±0.025	2.451±0.009
	LLMP Omit (Text)	MAE	0.750±0.051	0.807±0.044	0.770±0.077	0.738±0.074	0.722±0.056
		ACC	0.443±0.010	0.510±0.044	0.540±0.039	0.543±0.031	0.552±0.025
		NLL	2.641±0.073	2.465±0.042	2.465±0.044	2.393±0.039	2.400±0.018
	TabPFN	MAE	0.982±0.201	0.725±0.105	0.737±0.047	0.700±0.085	0.698±0.042
		ACC	0.302±0.080	0.412±0.063	0.448±0.068	0.472±0.040	0.468±0.005
		NLL	3.292±0.264	2.810±0.074	2.541±0.053	2.459±0.026	2.475±0.070
30	LLMP Impute (No Text)	MAE	0.958±0.074	0.956±0.016	0.907±0.053	0.903±0.030	0.884±0.072
		ACC	0.266±0.083	0.455±0.045	0.497±0.016	0.473±0.056	0.495±0.024
		NLL	3.515±0.441	2.888±0.067	2.684±0.072	2.679±0.053	2.648±0.082
	LLMP Omit (No Text)	MAE	0.967±0.067	1.014±0.056	0.976±0.082	0.953±0.039	0.959±0.059
		ACC	0.265±0.112	0.427±0.048	0.475±0.028	0.440±0.077	0.470±0.035
		NLL	4.010±0.222	3.691±0.315	3.521±0.321	3.420±0.223	3.307±0.209
	LLMP Impute (Text)	MAE	0.832±0.032	0.877±0.023	0.787±0.072	0.791±0.046	0.804±0.075
		ACC	0.437±0.035	0.528±0.047	0.522±0.046	0.523±0.033	0.508±0.007
		NLL	2.656±0.071	2.552±0.040	2.509±0.077	2.493±0.037	2.542±0.020
	LLMP Omit (Text)	MAE	0.804±0.034	0.860±0.033	0.805±0.058	0.790±0.037	0.750±0.074
		ACC	0.437±0.027	0.505±0.058	0.510±0.062	0.525±0.035	0.520±0.035
		NLL	2.730±0.042	2.581±0.015	2.569±0.011	2.513±0.029	2.486±0.016
	TabPFN	MAE	0.998±0.174	0.734±0.073	0.760±0.067	0.763±0.099	0.767±0.037
		ACC	0.292±0.070	0.425±0.030	0.440±0.024	0.475±0.016	0.462±0.019
		NLL	3.315±0.255	2.846±0.191	2.700±0.142	2.622±0.097	2.553±0.072
40	LLMP Impute (No Text)	MAE	0.966±0.040	0.976±0.042	0.956±0.077	0.931±0.026	0.928±0.077
		ACC	0.284±0.069	0.438±0.028	0.468±0.021	0.462±0.047	0.480±0.018
		NLL	3.475±0.311	2.948±0.019	2.782±0.110	2.750±0.055	2.718±0.134
	LLMP Omit (No Text)	MAE	0.974±0.020	1.050±0.050	1.012±0.058	1.024±0.015	1.000±0.056
		ACC	0.245±0.076	0.392±0.065	0.448±0.016	0.450±0.068	0.465±0.012
		NLL	4.229±0.595	3.777±0.251	3.656±0.244	3.563±0.161	3.539±0.193
	LLMP Impute (Text)	MAE	0.876±0.020	0.945±0.036	0.827±0.063	0.861±0.041	0.862±0.074
		ACC	0.440±0.017	0.512±0.033	0.525±0.020	0.523±0.016	0.507±0.007
		NLL	2.763±0.034	2.635±0.003	2.536±0.029	2.552±0.029	2.600±0.041
	LLMP Omit (Text)	MAE	0.865±0.017	0.930±0.028	0.846±0.028	0.838±0.013	0.803±0.068
		ACC	0.435±0.035	0.483±0.021	0.502±0.026	0.530±0.049	0.533±0.037
		NLL	2.836±0.018	2.678±0.005	2.628±0.016	2.596±0.032	2.551±0.048
	TabPFN	MAE	1.080±0.224	0.823±0.117	0.791±0.042	0.809±0.064	0.821±0.035
		ACC	0.268±0.102	0.423±0.045	0.438±0.019	0.490±0.014	0.457±0.024
		NLL	3.351±0.217	2.926±0.183	2.724±0.091	2.660±0.087	2.616±0.076

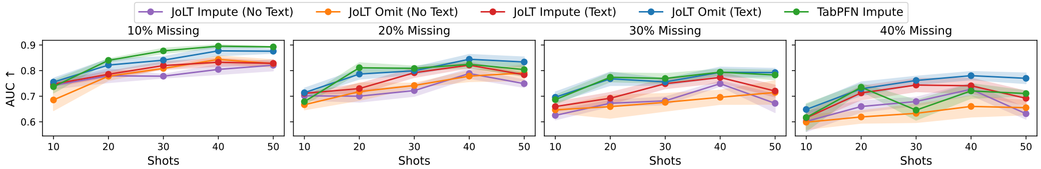

Figure F.3: AUC for JoLT using the gemma-2-27B LLM and TabPFN as a function of shots and percentage of data missing (MCAR) on the Car classification dataset (Bohanec, 1988). 200 test examples were used. The solid line and dots indicate the mean over 3 seeds which affect the training shot and test example selection as well as the missing pattern and the shaded region shows a confidence interval of one σ. Tabular version is in Table F.5

Table F.5: **Missing Data Handling on Cars**. AUC for JoLT using the gemma-2-27B LLM and TabPFN as a function of shots and percentage of data missing completely-at-random (MCAR) on the Car classification dataset (Bohanec, 1988). 200 test examples were used. Values are the mean and 95% confidence interval (CI) over 5 seeds that affect the training and test shot selection and the missing pattern.

		Shot				
Missing %	**Method**	**10**	**20**	**30**	**40**	**50**
10	JoLT Impute (No Text)	0.710±0.083	0.791±0.026	0.818±0.023	0.856±0.029	0.862±0.002
	JoLT Omit (No Text)	0.685±0.082	0.778±0.033	0.808±0.003	0.844±0.028	0.827±0.029
	JoLT Impute (Text)	0.736±0.061	0.813±0.010	0.819±0.012	0.851±0.022	0.856±0.007
	JoLT Omit (Text)	0.755±0.030	0.822±0.046	0.841±0.027	0.877±0.038	0.876±0.016
	TabPFN	0.737±0.049	0.840±0.012	0.877±0.021	0.895±0.011	0.892±0.004
20	JoLT Impute (No Text)	0.632±0.061	0.722±0.077	0.720±0.068	0.791±0.019	0.793±0.034
	JoLT Omit (No Text)	0.667±0.039	0.718±0.067	0.742±0.017	0.778±0.042	0.791±0.060
	JoLT Impute (Text)	0.678±0.042	0.787±0.037	0.779±0.028	0.824±0.018	0.819±0.022
	JoLT Omit (Text)	0.714±0.034	0.786±0.039	0.799±0.028	0.844±0.037	0.834±0.038
	TabPFN	0.679±0.024	0.811±0.038	0.808±0.018	0.824±0.029	0.804±0.048
30	JoLT Impute (No Text)	0.589±0.046	0.643±0.053	0.665±0.017	0.723±0.024	0.712±0.014
	JoLT Omit (No Text)	0.647±0.023	0.659±0.089	0.676±0.068	0.696±0.054	0.715±0.090
	JoLT Impute (Text)	0.614±0.058	0.722±0.037	0.716±0.015	0.749±0.017	0.745±0.030
	JoLT Omit (Text)	0.696±0.040	0.768±0.051	0.756±0.026	0.793±0.040	0.792±0.032
	TabPFN	0.687±0.034	0.774±0.035	0.769±0.037	0.794±0.024	0.783±0.036
40	JoLT Impute (No Text)	0.613±0.048	0.692±0.048	0.692±0.086	0.706±0.032	0.718±0.044
	JoLT Omit (No Text)	0.598±0.055	0.619±0.048	0.633±0.070	0.660±0.081	0.655±0.055
	JoLT Impute (Text)	0.670±0.039	0.737±0.029	0.733±0.038	0.763±0.035	0.758±0.030
	JoLT Omit (Text)	0.648±0.036	0.729±0.052	0.761±0.032	0.780±0.032	0.770±0.041
	TabPFN	0.617±0.105	0.735±0.028	0.646±0.077	0.721±0.056	0.710±0.021

F.4. Data Imputation

This section provides the tabular version of Fig. 5 in Table F.6. In summary, JoLT can outperform standard imputation techniques by leveraging text-based side information about the setting (i.e. the country name, the numerical columns contained medal counts for the 2024 Olympic games, etc.) that is not possible with numerical only methods.

Table F.6: **Imputation**: MAE as a function of % of missing data (MCAR) on the Paris 2024 Olympic Medals dataset for JoLT that uses the gemma-2-27B and four competive methods. Values are the mean and 95% confidence interval (CI) over 3 seeds that affect the training and test shot selection and the missing pattern.

	Missing %			
Method	**10**	**20**	**30**	**40**
Mean	3.398±1.190	3.523±0.784	3.835±0.762	3.796±0.540
kNN	1.728±0.505	1.833±0.226	2.291±0.636	2.279±0.604
Iterative	1.544±0.026	1.844±0.161	2.033±0.515	2.400±0.492
MICE	1.679±0.436	2.138±0.579	2.254±0.681	2.523±0.867
JoLT	1.139±0.253	1.590±0.306	1.465±0.044	1.543±0.337

Appendix G. Limitations

Along with the flexibility of LLMs, JoLT inherits their drawbacks. Maximum context sizes limit the size of tasks we can apply this method to and the amount of textual information we can condition on. JoLT is significantly more computationally expensive compared to competitive tabular prediction methods. In particular, both the computational complexity and the maximum context size of the LLM limit the number of training examples and the number of columns in the tabular dataset that can be reasonably processed. All of the experiments were performed on readily available small to medium sized open source LLMs that have fewer parameters and are generally less capable compared to large open source and proprietary LLMs that are accessed through services. We expect our results to improve and scale to larger tabular datasets with the use of proprietary LLMs.

Furthermore, we can combine ICL and fine-tuning to generalize to even larger datasets in the future.

Appendix H. Discussion

In this paper, we introduce JoLT, a novel method for probabilistic predictions on tabular data that leverages LLMs to define joint distributions over heterogeneous data types. JoLT distinguishes itself from competitive methods by effectively utilizing side information, providing joint probabilities, and automatically handling missing data, all without requiring additional training or data preprocessing. Through extensive experiments, we demonstrate that JoLT excels in low-shot scenarios, particularly when rich text-based side information is available, showcasing its versatility and practicality in real-world applications. Recently, Agarwal et al. (2024) demonstrated that in-context learning datasets with hundreds or thousands of shots yield performance benefits, indicating that scaling the number of shots might be an interesting direction for the future.

Impact Statement

Our work has demonstrated a new and useful ICL approach for generating probabilistic predictions on tabular data. It has the potential to allow practitioners from fields such as medical and ecological research to more easily employ probabilistic modeling and machine learning. Like all machine learning technology, there is potential for abuse, and possible consequences from incorrect predictions made with JoLT. Due to the black-box nature of the method, we do not know the biases in the underlying LLMs used and what effect they may have on JoLT output. However, LLM researchers are striving to make LLMs more fair and equitable. An open area of research is whether LLM biases propagate to JoLT predictions and whether de-biasing LLMs helps to fix such an issue.