# On The Role of Prompt Construction In Enhancing Efficacy and Efficiency of LLM-Based Tabular Data Generation

Banooqa Banday*[1], Kowshik Thopalli*[2], Tanzima Z. Islam[1], Jayaraman J. Thiagarajan [2]

[1]Texas State University
[2]Lawrence Livermore National Laboratory

***Abstract*—LLM-based data generation for real-world tabular data can be challenged by the lack of sufficient semantic context in feature names used to describe columns. We hypothesize that enriching prompts with even minimal contextual information, such as a brief explanation of what each feature represents can improve both the quality and efficiency of data generation. To test this, we investigate three prompt construction methods: Expert-guided, LLM-guided, and Novel-Mapping, with the latter two being automated approaches. Using the GReaT framework, our experiments show that context-enriched prompts significantly enhance the quality of the generated data while improving training efficiency. Notably, the LLM-guided method performed on par with expert-guided approaches, demonstrating its effectiveness as a scalable alternative.**

***Index Terms*—Large-language Models, Prompt Construction, Tabular Data, Synthetic Data Generation**

## I. Introduction

Generating realistic synthetic tabular data is a significant challenge with important applications in data augmentation [1], [2], privacy preservation [3], and data imputation [4]. While a wide variety of approaches have been proposed, LLM-based tabular data generation has emerged as a promising research direction. In this regard, Borisov *et al.* recently proposed GReaT (Generation of Realistic Tabular data) [5], which transforms tabular data into textual encodings (or prompts) and fine-tunes pre-trained LLMs to generate synthetic samples. More specifically, the text prompt in GReaT uses a subject-predicate-object schema, where the subject is the feature name. However, feature names in many real-world tabular datasets can be ambiguous, contain nondecipherable abbreviations or symbols, and even generic labels with no semantic context (e.g., *attribute A*, *attribute B*, etc.). In such cases, using these generic feature names might be insufficient to obtain high-fidelity synthetic samples.

In this paper, we hypothesize that enriching text prompts with domain-specific insights by explaining what each feature represents can significantly enhance an LLM's ability to synthesize high-quality tabular data. Specifically, we propose two automated prompt construction protocols (Figure 1) called `LLM-guided` and `Novel-Mapping`. For `LLM-guided`, if the feature names are partially specified, an external LLM is consulted to automatically expand feature descriptions based on the feature and dataset names. If the feature names are completely generic, then the `Novel-Mapping` approach provides an LLM with additional contextual information, such as value ranges and an arbitrary scientific domain name to improve the interpretation of feature names. We evaluate the effectiveness of this automated approach by comparing it to a method wherein a domain expert provides detailed feature descriptors (`Expert-guided`) to improve the interpretation of feature names.

Through experiments on diverse datasets and two different LLMs, we demonstrate that our context-enriched prompting strategies consistently outperform the baseline of using raw feature names, especially when the feature names are ambiguous or generic. The enhanced prompts not only improve the quality of the generated data, but also significantly boost training efficiency ($< 25\%$ of the epochs required by the baseline to achieve similar performance). Notably, the benefits persist even with parameter-efficient fine-tuning methods such as LoRA [6]. Our experiments also show that our proposed `LLM-guided` approach achieves just as good or better results automatically compared to the `Expert-guided` method. Given the high cost and scalability challenges of the expert-guided approach, the `LLM-guided` method offers a more efficient and scalable solution by eliminating the need for extensive manual input, making it a viable alternative for generating high-quality feature descriptors. Our key findings can be summarized as follows:

- When feature names are inherently interpretable, all approaches, including the baseline, demonstrate comparable performance in generating synthetic data.
- For datasets with cryptic or generic feature names, both `LLM-guided` and `Novel-Mapping` protocols offer significant improvements over the baseline method.
- The feature descriptions generated by the `LLM-guided` method can enable downstream analyses to achieve results that are comparable to or better than those from manually annotating feature names with the `Expert-guided` method.
- The benefits of our proposed methods persist even when using parameter-efficient fine-tuning techniques such as

* equal contribution

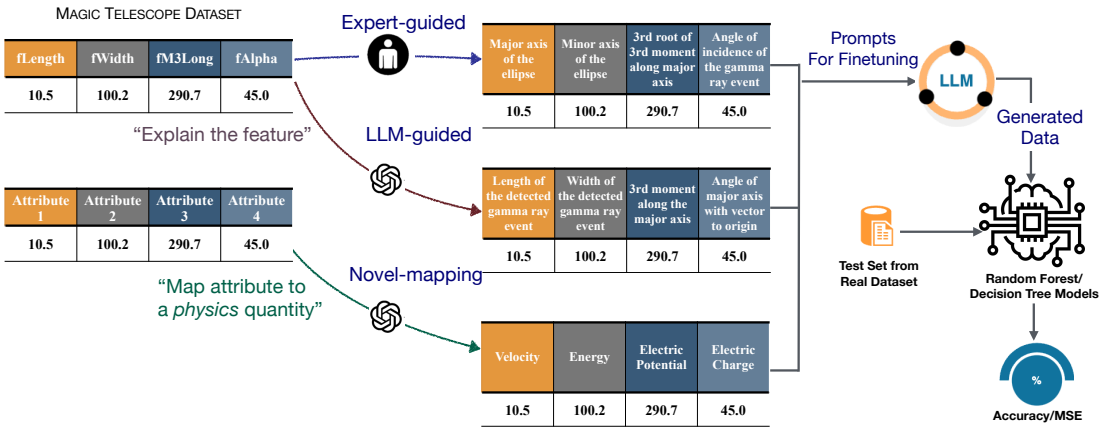


Fig. 1: An overview of our approach for LLM-based tabular data generation. Our contributions include designing new prompt construction strategies and investigating their role in improving the quality of synthesized samples.

LoRA, highlighting their robustness and versatility.

While we demonstrate the effectiveness of our proposed prompt construction method using GReaT, it's important to note that this methodology is generic and applicable to various prompting scenarios. Our approach differs from recent Retrieval Augmented Generation (RAG)-based techniques in its timing and integration of context. Unlike RAG, which incorporates context at runtime during the inference stage, our method integrates context at the input stage, before any learning occurs. This positions our approach as a preprocessing step that influences the LLM's learning process from the outset, complementing rather than replacing RAG-based knowledge augmentation techniques. The primary goal of this paper extends beyond improving isolated responses; we aim to study the influence of prompt enhancement on the generation of data distributions. By addressing the less-explored role of prompt engineering in shaping overall data patterns, our research fills a significant gap in the literature.

## II. Background

**Problem Setup**: Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote a tabular dataset with $N$ samples, where $\mathbf{x}_i \in \mathcal{X}$ is a set of features in $\mathbb{R}^n$ and $y_i \in \mathcal{Y}$ is the corresponding target (categorical or continuous-valued). Without loss of generality, we refer to the names of the $n$ input features in the table as $\{c_1 \cdots c_n\}$. We aim to build a generative model that can synthesize realistic samples $\{(\tilde{\mathbf{x}}_j, \tilde{y_j})\}_j$, where $\tilde{\mathbf{x}}_j \in \mathcal{X}$ and $\tilde{y_j} \in \mathcal{Y}$.

**An Overview of GReaT**: Our study builds on a tabular data generator called GReaT. In this approach, each $n-$dimensional input sample (a row in the input data matrix $\mathbf{X} \in \mathbb{R}^{T \times n}$) is first transformed into a textual encoding, and subsequently used as a prompt to query an LLM. This encoding strategy, which we refer to as `Baseline` encoding, constructs row-wise prompts by directly utilizing the feature names and adding the `is` qualifier to separate feature names and their corresponding values. E.g., the encoding for the $i^{\text{th}}$ row of the input matrix $\mathbf{X}$ can be written as "$c_1$ `is` $x_i^1, c_2$ `is` $x_i^2, \cdots, c_n$ `is` $x_i^n$", where $x_i^k$ represents the value of the $k^{\text{th}}$ feature from the $i^{\text{th}}$ sample. GReaT then fine-tunes pre-trained LLMs on these prompts using a next-token prediction objective. Once the model is fine-tuned, new samples can be unconditionally generated by post-processing the LLM's response for a test prompt that does not contain the feature values.

## III. Proposed Work

While the `Baseline` encoding has been shown to lead to strong generation capabilities [5], with real-world tabular datasets, the feature names are not always chosen to provide sufficient context for the observed values; for example, real-world datasets can contain ambiguous or generic feature names such as *Attribute A*. Furthermore, it is common to use abbreviations or symbols that are not readily decipherable (e.g., *fAlpha* in the magic telescope dataset) without sufficient expertise in the considered domain. In such cases, it can be challenging for an LLM to leverage useful priors from its pre-training, thereby impacting its generative performance. Consequently, the focus of this work is to study the impact of enriching the prompts with better clarity and specificity on an LLM's ability to generate high-quality tabular data. To this end, we propose a prompt construction protocol: `Expert-guided` to expand feature names with domain-specific descriptors during prompt construction. We also explore the effectiveness of providing further context such as value ranges and scientific domains to the LLM when expanding the feature names (`Novel-Mapping`). This paper compares the effectiveness of augmenting feature names with descriptions during prompt construction with that of manually adding domain-specific descriptions from expert knowledge (`Expert-guided`).

### A. Prompt Construction Protocols

**(i) `Expert-guided`**: This approach presents the best case scenario where an expert user is willing to put in manual efforts to describe the semantic context of each column in the tabular dataset while retaining the `is` qualifier from `Baseline` encoding.

**(ii) `LLM-guided`**: We propose to leverage a pre-trained Large Language Model (LLM) such as ChatGPT [7] to automate the feature-name expansion process. Specifically, we query the ChatGPT API with the following prompt: "For a dataset

named <name>, the given column names are <list of column names>. You need to provide a short one-line description of each feature." The response from ChatGPT is then parsed and used in lieu of the original feature names $c_k$ during prompt construction for the fine-tuning step. Note that, this approach is applicable only when the feature names are at least partially specified (e.g., abbreviations or symbols).

**(iii) `Novel-Mapping`**: Finally, in scenarios where the column names contain no useful information (e.g., Column A, Column B, $\cdots$), we propose the `Novel-Mapping` protocol, which will query an external LLM to generate a suitable feature name from an arbitrary domain (e.g., physics); for example, one can use the query "I have a dataset that does not have meaningful names for features. Given the ranges of the columns are <list of ranges>, suggest a term/phenomenon from <field name> that can take values in each of the given ranges. Rules are: (i) the terms/phenomenon should be from the same field, (ii) no two suggestions can be identical." Note, the <field name> can be arbitrarily chosen as long as the feature names remain consistent with the prior knowledge of the LLM (i.e., from the same domain) and have a similar range of feasible values (<list of ranges>). Figure 1 illustrates an example with the Magic Telescope dataset, where the LLM replaces the generic attribute labels with terms from physics.

### B. LLM Fine-tuning for Data Generation

While GReaT [5], by design, fine-tunes all LLM parameters, our study considers both regular fine-tuning as well as parameter-efficient fine-tuning (PEFT) based on LoRA [6] which is a technique for efficiently fine-tuning LLMs by restricting updates to a low-rank subspace of the model's gradient space, allowing significant parameter adaptation with minimal computational overhead.

### C. Implementation

For this study, we used two LLMs, namely GPT-2 [8] and DistilGPT-2 [9] and build upon the publicly released GReaT codebase [1]. Our implementation utilizes the Transformers [10] and PEFT [11] libraries [2]. We fine-tuned the LLMs using a 90-10 train-test split from all datasets, with training performed on the real-train split. For fine-tuning DistilGPT-2, we used the AdamW optimizer with learning rate $5e-5$ and trained for 400 epochs. For GPT-2, we used LoRA with learning rate set to $5e-5$, $r=16$ and $\alpha=32$.

TABLE I: Summary of datasets considered in this study.

| Dataset | Dataset Size | Features | Targets |
|---|---|---|---|
| HELOC [12] | 10459 | 23 | Likelihood of loan repayment (classification) |
| Magic Gamma Telescope [13] | 19020 | 10 | Class label – gamma ray or cosmic ray (classification) |
| California Housing [14] | 20640 | 8 | Median house value (regression) |
| Parkinson's Diagnosis [15] | 5875 | 19 | Parkinson's score (regression) |

[1]https://github.com/kathrinse/be_great

[2]https://github.com/huggingface/

TABLE II: Prediction performance of decision tree and random forest models on four datasets. ML models are trained on data generated by fine-tuning Distil-GPT2. Results demonstrate that enriching prompts with relevant semantic context yields a boost in performance.

| Dataset (Metric) | Prompting Protocol | Performance | |
|---|---|---|---|
| | | Decision Tree | Random Forest |
| Magic Telescope (Accuracy) | `Baseline` | 80.57 | 82.94 |
| | `Expert-guided` | **82.1** | **86.25** |
| | `LLM-guided` | 80.6 | 83.81 |
| HELOC (Accuracy) | `Baseline` | 69.12 | 70.65 |
| | `Expert-guided` | 69.22 | **70.7** |
| | `LLM-guided` | **69.36** | 70.27 |
| Parkinsons Diagnosis (MSE) | `Baseline` | 11.15 | 10.2 |
| | `Expert-guided` | 4.21 | 1.96 |
| | `LLM-guided` | **3.52** | **1.84** |
| California Housing (MSE) | `Baseline` | 0.5 | 0.35 |
| | `Expert-guided` | **0.46** | **0.34** |
| | `LLM-guided` | 0.48 | **0.34** |

## IV. Experimental Setup

**Datasets.** Table I summarizes the datasets used in this work.

**Evaluation.** To assess the quality of the synthetic data generated with our prompting strategies, we test how well predictive models trained solely on this data perform on real test data. In prior work, this evaluation has been referred to as machine learning efficiency (MLE) [5]. To estimate MLE, we utilize two widely used ML models for tabular data – random forests (RF) [16] and decision trees (DT) [17]. We train these models using the sklearn [18] library and conduct hyper-parameter tuning through grid-search with 5-fold cross-validation. As evaluation metrics, we use the mean squared error and accuracy scores for regression and classification tasks respectively. MLE quantifies how well models trained purely on the synthetic data can generalize to real unseen data, thereby serving as an effective proxy for the quality of the generated samples. Note, that we fine-tune LLMs with real-train split and evaluate MLE on real-test split.

## V. Results and Findings

In this section, we perform a comprehensive evaluation of the impact of the prompt construction methods on MLE. We present our findings below

**Finding 1:** *Leveraging semantic context in prompts boosts LLM-based data generation.*

Table II presents the MLE scores of models trained on synthetic data generated using various prompting methods, after fine-tuning all DistilGPT-2 parameters. For the Parkinsons dataset, the `LLM-guided` prompts reduces prediction error by up to 68% over the baseline and outperforms the manually generated `Expert-guided` prompts by 16%.

**Finding 2:** *Better prompts improve training efficiency.*

Figure 2 provides insights into the training dynamics when using the proposed prompting strategies on the Parkinson's diagnosis dataset. Strikingly, with both `LLM-guided` and `Expert-guided` prompts, the models surpass the MLE while requiring $< 25\%$ of the `Baseline` training epochs.

On the Parkinson's diagnosis dataset, the expert-guided and LLM-guided approaches reduce the MSE by $> 80\%$ compared

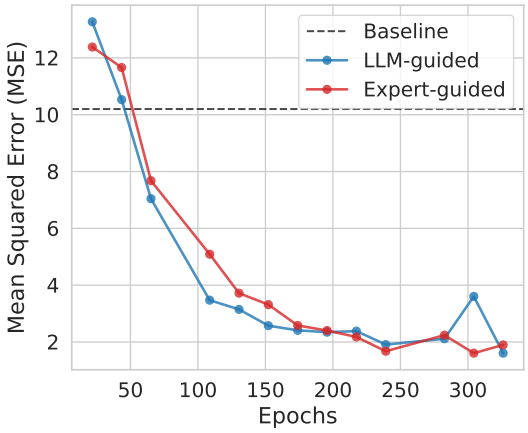


Fig. 2: Enhanced prompt construction strategies lead to better computational efficiency.

to the `Baseline` encoding for both ML models. Enhanced prompts do not provide significant performance gains on the HELOC and California Housing datasets, which already contain non-ambiguous and readily interpretable feature names.

**Finding 3:** *Benefits continue to persist even with parameter-efficient fine-tuning*.

Figure 3 presents the MLE scores achieved by models trained on synthetic data generated from the GPT-2 model fine-tuned with LoRA.

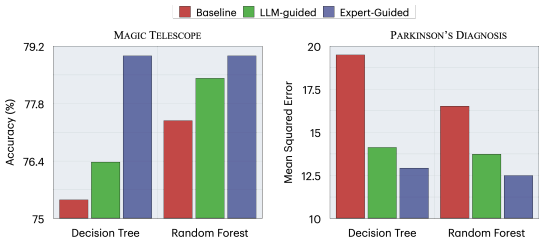


Fig. 3: Performance of ML models trained on synthetic data, generated by fine-tuning GPT-2 with LoRA using various prompting methods, evaluated on the Magic Telescope and Parkinson's diagnosis datasets.

A striking observation is that the proposed prompting strategies continue to outperform `Baseline` encoding even with PEFT. For example, on the Parkinson's diagnosis dataset, the `Expert-guided` and `LLM-guided` prompts reduce the MSE by 33.7% and 27.5%, respectively, compared to `Baseline` prompting for the DT model. Furthermore, on the Magic Telescope dataset, `Expert-guided` prompts achieve a non-trivial accuracy boost of 3.99%.

**Finding 4:** *When no context is available*, `Novel-Mapping` *is highly effective*.

In Figure 4, we present the downstream prediction performance obtained with the `Baseline` and the `Novel-Mapping` strategies. Notably, when dealing with datasets containing non-decipherable names and regression tasks, mapping those feature names to meaningful ones from another domain that is consistent with the priors of the pre-trained LLM provides benefits. For instance, in the case of the Magic Telescope dataset, we observe an accuracy improvement of 1.5% for the DT model. Similarly, for Parkinson's diagnosis, we observe substantial reductions of >57% in MSE. We only compare the `Novel-Mapping` strategy with `Baseline` as the `LLM-guided` method would otherwise produce random descriptions and no expert knowledge is available for that dataset.

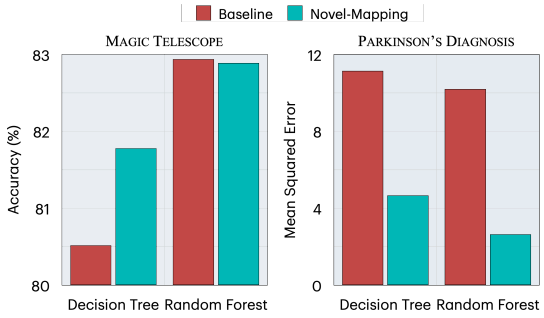


Fig. 4: Mapping generic feature names to semantically meaningful descriptors from a novel domain provides non-trivial gains in performance.

## VI. Conclusions

Our empirical results show that when the feature names in tabular datasets do provide sufficient semantic context, the proposed prompting strategies can substantially enhance the quality of the generated samples. Furthermore, these strategies also exhibit improved computational efficiency. Interestingly, even `Novel-Mapping` is a viable strategy in practice, particularly when the dataset contains only generic attribute descriptors. Additionally, we compute distance to closest record (DCR) that has been used before [5] to ensure that the generated data do not contain copy of real data points. Augmenting real training data with these synthetic samples improved the accuracy of the decision tree by 1. 1% in the Magic Telescope dataset and by $\sim 3\%$ for the Heloc dataset.

## VII. Limitations

We highlight some of the limitations of our approach which warrant further investigation. First, while we considered a diverse set of datasets, we only focus on four of them, and considering a more diverse range of datasets is required. Second, we primarily assess the quality of the generated data using the Machine Learning Efficiency (MLE) metric, which need not capture all aspects of data quality. Finally, while we propose the `LLM-guided` and `Novel-Mapping` strategies to address the limitation of relying on human expertise (`Expert-guided` approach), further research is needed to validate their effectiveness across a wider range of scenarios.

## Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science under Award Number DE-SC0022843. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, Lawrence Livermore National Security, LLC. This work is supported by LDRD project 24-FS-002.

# REFERENCES


[1] Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using llms: Data perspectives, learning paradigms and challenges. *arXiv preprint arXiv:2403.02990*, 2024.

[2] Jiayi Yuan, Ruixiang Tang, Xiaoqian Jiang, and Xia Hu. Large language models for healthcare data augmentation: An example on patient-trial matching. In *AMIA Annual Symposium Proceedings*, volume 2023, page 1324. American Medical Informatics Association, 2023.

[3] Mohammad Al-Rubaie and J. Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

[4] Xue Li and Till Döhmen. Towards efficient data wrangling with llms using code generation. In *DEEM@ SIGMOD*, pages 62–66, 2024.

[5] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

[6] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[7] OpenAI. Chatgpt (gpt-4). `https://www.openai.com/`, 2024. https://www.openai.com/.

[8] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[10] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[11] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

[12] Kaggle. Home equity line of credit(heloc).

[13] R. Bock. MAGIC Gamma Telescope. UCI Machine Learning Repository, 2007. DOI: https://doi.org/10.24432/C52C8B.

[14] Kaggle. California housing prices.

[15] Athanasios Tsanas and Max Little. Parkinsons Telemonitoring. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C5ZS3N.

[16] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[17] Leo Breiman, Jerome Friedman, R.A. Olshen, and Charles J. Stone. Classification and regression trees (1st ed.). Chapman and Hall/CRC, 1984. DOI: https://doi.org/10.1201/9781315139470.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.