DataVisT5: A Pre-trained Language Model for Jointly Understanding Text and Data Visualization

Zhuoyue Wan*, Yuanfeng Song†, Shuaimin Li*, Chen Jason Zhang*, Raymond Chi-Wing Wong‡
*PolyU, Hong Kong, China †WeBank Co., Ltd, Shenzhen, China ‡HKUST, Hong Kong, China

***Abstract*—Data visualization (DV) is the fundamental and premise tool to improve the efficiency in conveying the insights behind the big data, which has been widely accepted in existing data-driven world. Task automation in DV, such as converting natural language queries to visualizations (i.e., text-to-vis), generating explanations from visualizations (i.e., vis-to-text), answering DV-related questions in free form (i.e. FeVisQA), and explicating tabular data (i.e., table-to-text), is vital for advancing the field. Despite their potential, the application of pre-trained language models (PLMs) like T5 and BERT in DV has been limited by high costs and challenges in handling cross-modal information, leading to few studies on PLMs for DV. We introduce DataVisT5, a novel PLM tailored for DV that enhances the T5 architecture through a hybrid objective pre-training and multi-task fine-tuning strategy, integrating text and DV datasets to effectively interpret cross-modal semantics. Extensive evaluations on public datasets show that DataVisT5 consistently outperforms current state-of-the-art models and higher-parameter Large Language Models (LLMs) on various DV-related tasks. We anticipate that DataVisT5 will not only inspire further research on vertical PLMs but also expand the range of applications for PLMs.**

***Index Terms*—pre-trained language model, data visualization, text-to-vis, vis-to-text, FeVisQA, table-to-text**

I. Introduction

Data visualizations (DVs) utilize graphical representation to convey insights to summarize the massive raw data, which is a common practice in existing big data era [1], [2]. Popular data analysis and database applications, such as Google Sheets[1] and Microsoft Power BI[2], all support DV features. Many institutions realize the value of DV and have applied it as their daily fundamental tools. Thus the ability of creating suitable DVs has become a necessary skill for data analysts, engineers, and data scientists [3]–[5]. However, creating appropriate DVs remains challenging, even for experts, since it requires visual analysis expertise and familiarity with the domain data. Furthermore, users must master the complex grammar of *Declarative Visualization Languages* (DVLs), such as Vega-Lite [6], ggplot2 [7], and Vega-Zero [8], to accurately define DV *specification* in the visualization engine.

To lower the barriers to creating DVs and further unlock the power of DV for the general public, researchers have proposed a variety of DV-related tasks that have attracted significant attention from both industrial and academic researchers. These tasks include *text-to-vis* (i.e., automatically generating DVs from natural language questions) [8], [9], *vis-to-text* [10] (i.e., automatically generating interpretations of complex DVs for educational purposes), *FeVisQA* [11] (i.e., free-form question answering over data visualization), and *table-to-text* (i.e., describing a given table) [12].

A vivid example is given in Figure 1, which shows four important tasks central to the domain knowledge of DV: *text-to-vis*, *vis-to-text*, *FeVisQA* and *table-to-text*. The figure presents a natural language (NL) question, *"Give me a pie chart about the proportion of the number of countries in the artist table."* This example demonstrates the text-to-vis task's capability to interpret the NL question and transform it into a Vega-Lite specification, resulting in a pie chart. The *DV query*, introduced by [9], serves as a bridge in the text-to-vis process, encapsulating visualization details and data operations with a grammar akin to SQL. Translations between DV queries and DVLs are seamless, with text-to-vis tasks primarily focusing on converting NL questions into DV queries. Conversely, the vis-to-text task aims to generate accessible and user-friendly explanations of complex visualizations for individuals without expertise in the field. The FeVisQA task addresses user inquiries regarding DV by providing detailed answers to common questions. We present four typical DV-related questions, including understanding the semantics of a DV query, resolving numerical issues within a chart, and evaluating the compatibility of a DV query with a given database. Lastly, the table-to-text task generates informative NL description of tabular data, which are essential for visual analytics, thereby reducing the perceptual effort needed for data interpretation.

Meanwhile, PLMs such as BERT [13] and T5 [14] have received considerable attention in the realms of natural language processing (NLP) and data mining, becoming widely recognized for their efficacy. These PLMs greatly promote the development of effective text-driven applications, since they show dominating performance in understanding the semantics in natural language. The operational paradigm for these PLMs typically unfolds in two stages: initially, they undergo unsupervised pre-training on expansive, open-domain datasets (such as Wikipedia) to acquire foundational capabilities in language representation and comprehension; subsequently, they are fine-tuned on specialized corpora pertinent to targeted downstream tasks, thereby enhancing task-specific performance. Despite their success [15]–[17], there are still significant challenges when it comes to the DV field : (i) Limited studies have been conducted to explore the effectiveness of PLMs in capturing the DV semantics. (ii) Since there is a substantial modal gap between the DV modality and the text modality, satisfied performances cannot be achieved by directly applying

[1]https://www.google.com/sheets/about/
[2]https://powerbi.microsoft.com/

2375-026X/25/$31.00 ©2025 IEEE
DOI 10.1109/ICDE65448.2025.00131

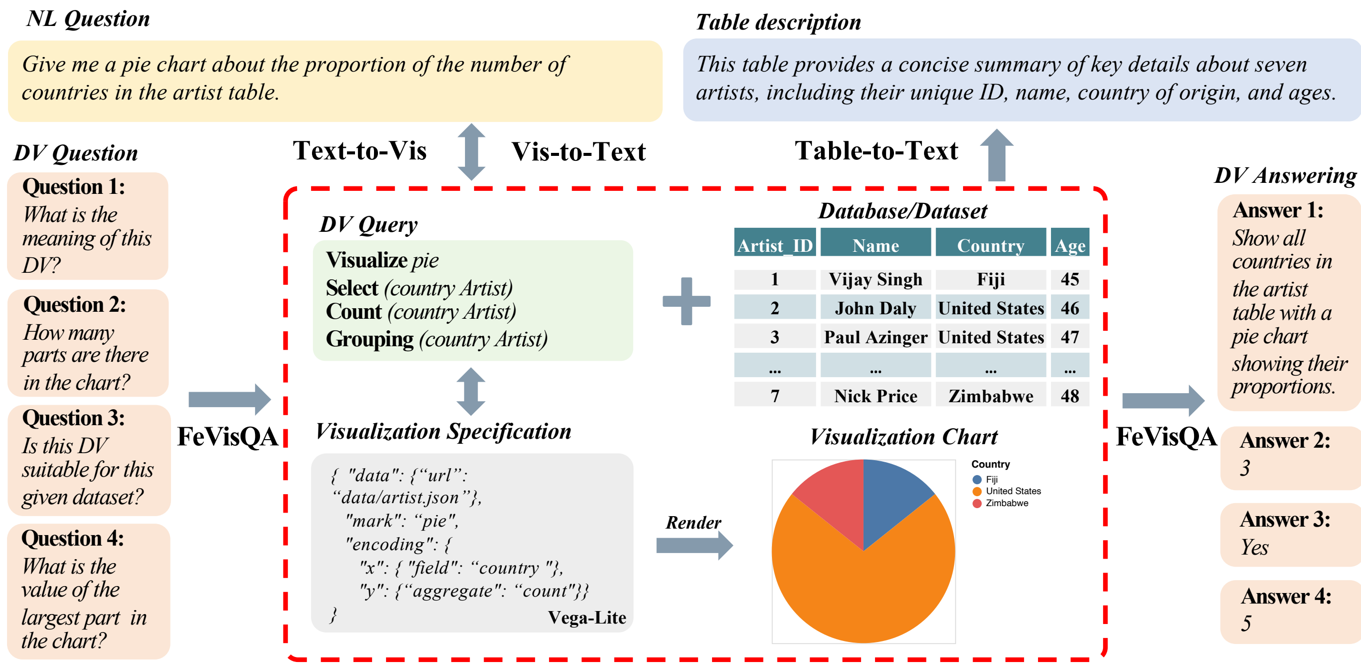


Fig. 1: An illustration depicting the *text-to-vis*, *vis-to-text*, *table-to-text*, and *free-form question-answering over data visualization* problems, showcasing examples including a NL question, a DV query, a DVL visualization specification, a table description, a visualization chart, and four question-answer pairs.

existing PLMs (e.g., T5) to DV-related tasks mentioned above. (iii) In the DV area, a possible PLM needs the ability of handling cross-modal information (i.e., text and DV), while also being capable of managing multiple distinct tasks.

To alleviate above-mentioned problems, we propose a novel PLM for jointly understanding text and DV, refereed as **DataVisT5** in this paper. Based on text-centric T5 architecture, we enhance the pre-training process by incorporating a comprehensive array of cross-modal datasets that integrate natural language with DV knowledge, encompassing DV queries, database schemas, and tables. Since DV queries resemble programming language-like queries, we employ CodeT5+ [18] as the starting checkpoint in our work. This choice leverages the robust code semantic understanding and generation capabilities of CodeT5+, providing DataVisT5 with a substantial advantage in generating and comprehending the unique programming language of our DV tasks. Building on this foundation, we apply *table-level database schema filtration* to reduce training complexity. Addressing the challenges of format consistency between DV and textual modalities, we introduce a *unified encoding format* for DV knowledge that facilitates the convergence of text and DV modalities. To eliminate stylistic discrepancies in manually curated datasets, we adopt standardized encoding.

Additionally, the pre-training objectives for DataVisT5 are twofold: (i) the span corruption approach of Masked Language Modeling as utilized by the original T5 model, and (ii) a Bidirectional Dual-Corpus objective that operates on source-target pairings. After the mixed-objective pre-training, we conduct multi-task fine-tuning (MFT) of our DataVisT5 on DV-related tasks including text-to-vis, vis-to-text, FeVisQA, and table-to-text. To substantiate the rationale behind our proposed model, we performed comprehensive experimental evaluations on various public datasets. The results consistently demonstrate that DataVisT5 surpasses the state-of-the-art (SOTA) models and higher-parameter LLMs. In summary, our main contributions are as follows:

- We introduce and release DataVisT5: the first Pre-trained Language Model (PLM) tailored for the joint understanding of text and DV. This innovation opens avenues for future research on task-specific PLMs and enriches the landscape of PLM designs.
- We enhance the text-centric T5 architecture to handle cross-modal information. Our novel hybrid pre-training objectives are conceived to unravel the complex interplay between DV and textual data, fostering a deeper integration of cross-modal insights.
- Extensive experiments on public datasets for diverse DV tasks including text-to-vis, vis-to-text, FeVisQA, and table-to-text demonstrate that DataVisT5 excels in multi-task settings, consistently outperforming strong baselines and establishing new SOTA performances.

II. Preliminary

This section provides the foundational concepts and definitions pivotal to DV-related tasks, with the objective of cultivating a more profound understanding.

Natural Language Question. An NL question enables users, even those with a minimal background in DV and programming skills, to formulate queries intuitively. Figure 1 demonstrates such an instance, with the user's request articulated as, "*Give* *me a pie chart about the proportion of the number of countries in the artist table*".

Declarative Visualization Language. Transforming data into a graphical representation typically involves the use of a declarative visualization language (DVL). This kind of language provides a set of specifications that determine the construction of visualizations. These specifications include various elements such as chart type, colors, sizes, and mapping functions, as well as properties for visual marks like canvas dimensions and legends. Several DVLs are prevalent in the field, such as Vega-Lite [6], ggplot2 [7], ZQL [19], ECharts [20], Vega-Zero [8], and VizQL [21], each offering unique features to facilitate the visualization process.

Visualization Specification. A visualization specification comprises a *JSON* format object that delineates the dataset and its visual attributes (such as chart types and data transformation functions) in accordance with the syntax of a specific DVL. It is noteworthy that each DVL possesses a unique grammar, necessitating distinct visualization specifications for rendering the same DV chart.

Data Visualization Query. Introduced by [22], [23], a framework for querying a database for visual data representations seeks to encapsulate the full spectrum of potential DVLs. As depicted in Figure 1, a DV query specifies a "pie" chart and integrates SQL-like operations (e.g. *Count* and *Order By*). This versatile DV query format can be converted into visualization specifications for different DVLs, enabling visualization engines to render the specified chart.

Data Visualization Chart. The DV charts are the visual representations such as scatters, bars, or maps used to convey the data summary and insights defined by the visualization specification. In Figure 1, the final visualization result is the bar chart that corresponds to the NL question.

III. Our Proposed Model: DataVisT5

We present our proposed DataVisT5 model, with the pipeline overview in Section III-A. This is followed by details on database schema filtration in Section III-B, DV knowledge encoding in Section III-C, and standardized encoding in Section III-D. We discuss our hybrid pre-training objectives in Section III-E and conclude with our multi-task fine-tuning strategy in Section III-F.

A. Pipeline Overview

Figure 2 provides an overview of the complete pipeline, comprising five main stages: (1) *Database schema filtration*, (2) *DV knowledge Encoding*, (3) *Standardized Encoding*, (4)*Model Pre-training*, and (5) *Model Fine-tuning*. The *Database schema filtration* process involves comparing n-grams extracted from the given database schema with those present in the text under consideration, enabling us to identify referenced tables in the question and acquire a sub-database schema that aligns semantically. During the *DV knowledge Encoding* phase, we linearize DV knowledge encompassing DV queries, database schemas, and tables. Subsequently, in the *Standardized Encoding* phase, we normalize the DV knowledge to facilitate more efficient learning. The resulting corpus, in its unified form, is then employed to train our proposed DataVisT5 model.

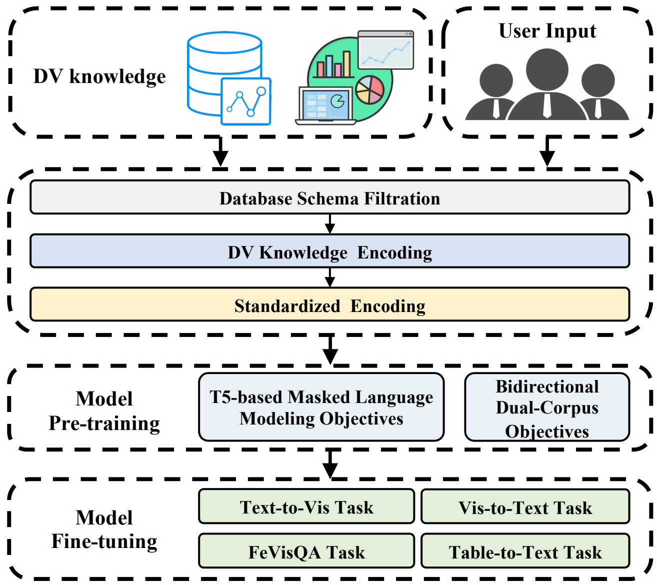


Fig. 2: The pipeline of DataVisT5.

B. Database Schema Filtration

Before the integration of DV and text modalities, it is critical to recognize that NL questions can incorporate keywords related to the database schema. This requires the explicit identification of references to columns, tables, and conditional values within the NL questions. To address this challenge, we employ N-gram matching as a method due to its simplicity of implementation and notable effectiveness for a variety of applications. In an effort to minimize information loss, our primary focus is at the table level, where we compare N-grams extracted from the NL questions to those present in the database tables. Following the initial comparison, we refine the obtained sub-schema by considering the implicated tables and their respective columns.

C. DV Knowledge Encoding

To address the disparity between text and DV modalities, we propose investigating unified formats for DV knowledge. The connection between natural language and DV knowledge poses challenges due to limited data accessibility. Nevertheless, a unified format allows models to capitalize on extensive pretraining for smaller datasets. Employing consistent formatting, as recommended by [24], offers advantages in multi-task training and mitigates performance decline caused by data heterogeneity compared to single-task training. The subsequent sections provide a comprehensive introduction to the unified representation of three distinct types of DV knowledge: DV queries, database schemas, and tables.

Encoding DV query. While most existing NLP models, such as [13], consider NL inputs as flat text sequences, we adopt a similar approach for modeling a DV query by treating it as a plain text sequence in a straightforward manner.

Encoding Database schema. The database schema comprises tables and columns. For each table in the schema, the table

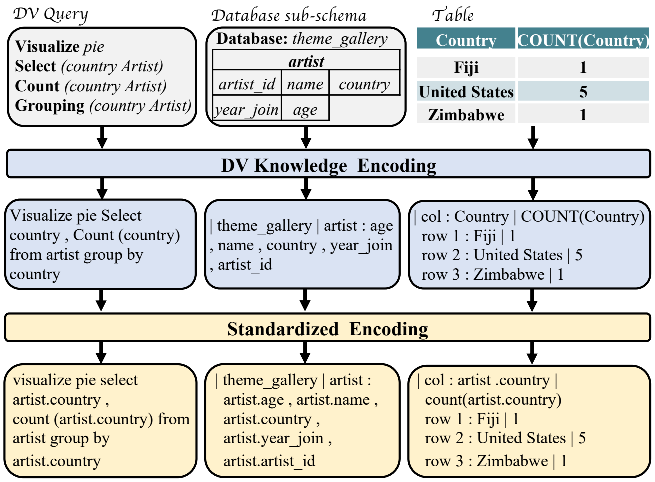


Fig. 3: Examples of DV Knowledge Encoding and Standardized Encoding from NVBench.

name is followed by a list of its columns formatted as "$table$: $column_1$, ... $column_n$". Different tables are joined using the symbol "|". Additionally, the database name is prefixed to the generated sequence with boundaries indicated by "|".

Encoding Table. Following [25], we employ a sequential representation of tables, akin to the schema encoding technique, which uses distinctive tokens to delineate table structure. The table is linearly represented as "$col : c_1 \mid \cdots \mid c_N \; row\ 1 : v_{11} \mid \cdots \mid v_{1N} \; \cdots \; row\ M : v_{M1} \mid \cdots \mid v_{MN}$", with N indicating the total column count and M representing the row count.

Example. An presented in Figure 3, where (1) the DV query is sequentially encoded into text sequences based on the data manipulation operations: *Visualize*, *Select*, *Count*, and *Grouping*, (2) the filtered database sub-schema, including the database name (*theme_gallery*), table name (*artist*), and columns, is encoded into a corresponding text sequence, and (3) the table content is linearly encoded in the format "*col: Country* | *COUNT(Country)*", along with the remaining three rows of the table.

D. Standardized Encoding

Due to the manual generation of queries by multiple annotators with diverse annotation habits, subtle stylistic differences are prevalent in the final annotated DV queries within NVbench, including variations in the capitalization of keywords. Similar to issues encountered with SQL queries, these stylistic inconsistencies, while not affecting the model's execution results, pose an additional learning challenge that must be addressed. To address the stylistic variations in DV queries, a preprocessing strategy was implemented before training. This strategy includes: (1) affixing the primary table name *T* to the selected columns *col*, resulting in the notation *T.col* across DV queries; particularly, for instances where the wildcard symbol * is employed in a *COUNT* function, *COUNT(*)* is replaced with *COUNT(T.col)* to maintain uniformity; (2) the insertion of spaces surrounding parentheses and the replacement of double quotes with single quotes; (3) the inclusion of the *ASC* keyword subsequent to the *ORDER BY* clause when ordering is not explicitly specified; (4) the elimination of the *AS* clause and the substitution of table aliases (e.g., *T1*, *T2*) with their actual table names; (5) the lowercase conversion.

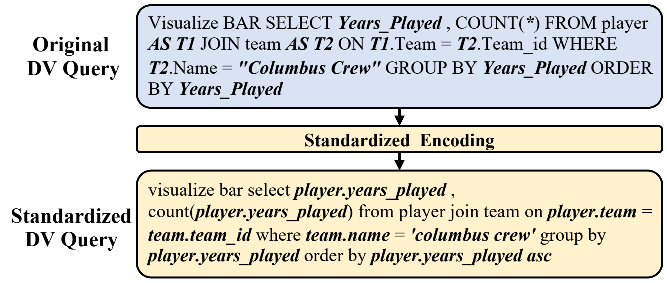


Fig. 4: An Standardized DV Query with join operation example.

Example. In a DV query with a $Join$ operation, as depicted in Figure 4, standardization involves renaming table aliases $T1$ and $T2$ to $player$ and $team$, respectively. The query's $COUNT(*)$ is specified as $COUNT(player.years\_played)$, 'Columbus Crew' is quoted with single quotes, the ASC keyword is appended if sort order is absent, and the entire query is cast to lowercase.

In alignment with the standardization of DV queries, similar encoding steps are applied to database schemas and tables to ensure consistency. This includes affixing the table name T to each column name col and converting them to $T.col$.

E. Hybrid Pre-training Objectives

Bidirectional Dual-Corpus Objectives. To address divergence between the pretraining and fine-tuning phases, we introduce Bidirectional Dual-Corpus (BDC) objectives during pretraining.In this approach, both the source and target corpora are randomly selected with equal probability (0.5) during model training to serve as the input. The remaining corpus is then used as the output for translation purposes. Accordingly, for a target sequence of T tokens, we define the BDC loss function, $\mathcal{L}_{BDC}(\theta)$, as follows:

$$\mathcal{L}_{BDC}(\theta) = \sum_{i=1}^{T} - \log P_\theta(t_i \mid \mathbf{s}, \mathbf{t}_{<i}), \tag{1}$$

where $\mathbf{s}$ signifies the source input, $\mathbf{t}_{<i}$ represents the sequence of tokens generated by the decoder up to but not including the i-th token, and t_i is the token that the decoder is tasked with predicting. The term θ denotes the model parameters.

As depicted in Figure 5, the segment highlighted by arrows elucidates the deployment of the BDC Objectives, encompassing four discrete tasks germane to DV. A comprehensive definition of these tasks is deferred to Section V. To enhance task-specific processing and facilitate knowledge transfer across different modalities, we introduce unique special tokens. For example, as demonstrated in Figure 5, the Text-to-Vis task utilizes a special token $<NL>$ to prefix the NL question corpus and $<VQL>$ for the DV query corpus. In contrast, for the FeVisQA task, DV question-answer pairings are delineated with the tokens $<Question>$ and $<Answer>$ to signify their respective components.

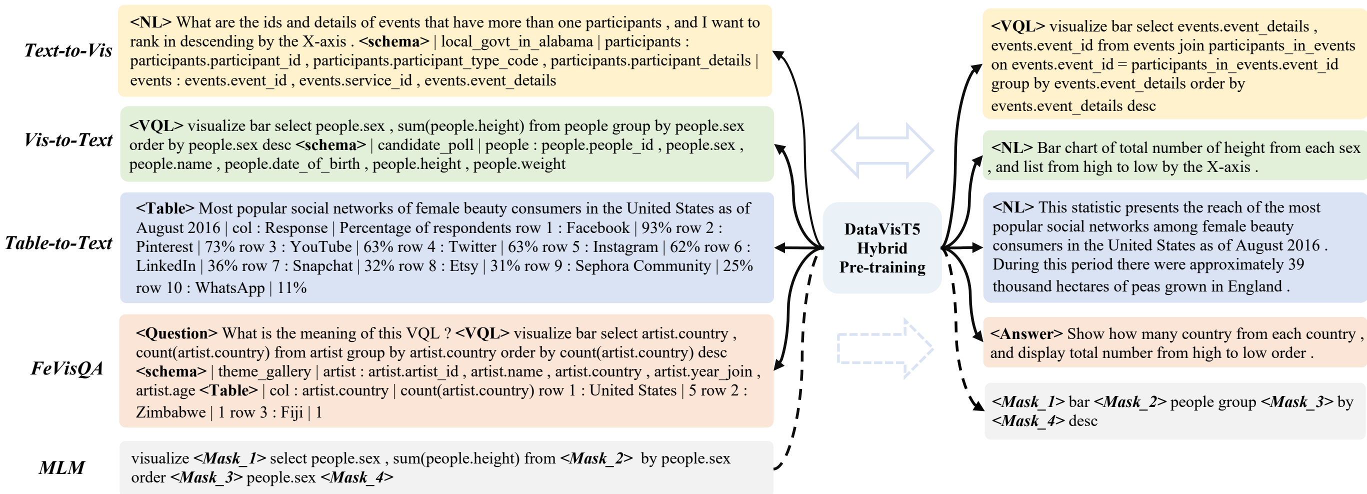


Fig. 5: Overview of hybrid pre-training objectives. The solid lines denote the Bidirectional Dual-Corpus objectives, which facilitate the learning of language representation by leveraging bidirectional context. The dashed lines represent the T5-based MLM objectives, designed to reconstruct the original input from masked tokens.

T5-based MLM Objectives. The application of Masked Language Modeling (MLM) as a pretraining objective is pivotal for pretraining encoder-decoder models. In our study, we employed the span corruption MLM strategy from [14], where consecutive words in the input are replaced by sentinel tokens, and the decoder generates the omitted text, each instance preceded by its respective sentinel. To ensure consistency with the pretraining checkpoint, we maintained an average span length of 3 subword tokens across the input corpus and masked 15% of the subwords. This MLM objective was applied to a cross-modal corpus comprising text, DV query, database schema, and table. Over a sequence of N tokens, our T5-based MLM loss is defined as:

$$\mathcal{L}_{MLM}(\theta) = \sum_{n=1}^{N} - \log P_\theta \left(x_n^{\mathrm{m}} \mid \mathbf{x}^{\setminus \mathrm{m}}, \mathbf{x}_{<n}^{\mathrm{m}} \right), \tag{2}$$

where θ are the model parameters, x_n^{m} is the masked token predicted by the decoder, $\mathbf{x}^{\setminus \mathrm{m}}$ represents the unmasked encoded inputs, and $\mathbf{x}_{<n}^{\mathrm{m}}$ is the sequence of tokens generated by the decoder up to but not including the n-th token.

An illustration is presented in Figure 5, where the segments linked by dashed lines pertain to the T5-based MLM Objectives. This figure showcases the application of span denoising targets to a DV query. Within this query, the terms "bar", "people group", "by", and "desc" are selected at random. Subsequently, a subset of these terms is replaced by sentinel tokens, illustrated as <Mask_1>, <Mask_2>, <Mask_3>, and <Mask_4>.

Hybrid Objectives. After achieving the aforementioned two objectives, we create a hybrid objective by sampling from both the MLM Objectives and the BDC Objectives corpora. Consequently, each training mini-batch is composed of examples drawn from a cross-modal corpus, each formatted to align with diverse learning objectives. We adopt a final hybrid loss $\mathcal{L}_H$:

$$\mathcal{L}_H(\theta) = \mathcal{L}_{BDC}(\theta) + \mathcal{L}_{MLM}(\theta), \tag{3}$$

which enables DataVisT5's readiness for multiple DV-related downstream tasks demanding contextual comprehension and pattern recognition.

F. Multi-Task Fine-tuning

To achieve better performance in multiple downstream tasks related to DataVisT5, we employ temperature mixing to combine the training data of all tasks. The temperature value is set to 2, following [14]. Temperature up-sampling helps balance the influence of each task on the model by adjusting the probability of selecting data from each task during training. This prevents larger datasets from overpowering smaller ones. By merging training data from different tasks, the model is encouraged to learn representations that are beneficial across various corpora. Consequently, this leads to improved generalization and a more robust model capable of handling diverse DV tasks.

IV. Pretraining Dataset Construction

We have constructed a dataset tailored for our Hybrid Pretraining Objectives by integrating four public datasets. The following sections outline our pretraining dataset construction, detailing data collection in Section IV-A, data processing in Section IV-B, and data partitioning in Section IV-C.

A. Data Collection

1) NVBench: The NVBench dataset [9] represents a publicly accessible NL2Vis corpus, containing 7,219 pairs of NL questions and their corresponding DV queries. It was originally curated to evaluate the efficacy of models in transforming textual queries into visual representations. As the most commonly utilized dataset in this domain, NVBench has been employed in several prominent studies, including those by [8], [10], [26] Table I offers a detailed overview of the NVBench dataset, comprising 25,628 entries that have been collated from 152 distinct databases originating from the Spider dataset [27]. To facilitate fair comparison with other established baselines as

TABLE I: The statistics of the NVBench dataset

	Number of instances		Number of databases	
Split	NVBench w/o join	NVBench	NVBench w/o join	NVBench
Train	10564	16780	98	106
Valid	2539	3505	15	16
Test	2661	5343	27	30
Total	15764	25628	140	152

TABLE II: The statistics of the Chart2text and WikiTableText datasets

	Number of instances			Number of cells	
Split	Chart2Text	WikiTableText	**Metrics**	Chart2Text	WikiTableText
Train	24368	10000	Min.	4	27
Valid	5222	1318	Max.	8000	108
Test	5221	2000	$\leq$150	34272	13318
Total	34811	13318	$>$150	539	0

discussed in Section V, we meticulously separated the DV queries involving non-join operations from those that include join operations and performed an in-depth statistical analysis. Specifically, the dataset contains 15,764 samples without join operations. DV queries that employ non-join operations, utilizing a single table, are showcased in Figure 3. Conversely, DV queries featuring join operations, where multiple tables are engaged, are illustrated in Figure 4.

2) Chart2text.: The chart-to-text conversion process, as introduced by [28], constitutes a comprehensive benchmark incorporating two distinct datasets, cumulatively consisting of 44,096 charts that span an extensive array of subjects and graphical representations. The data for this benchmark originates from two primary sources: Statista[3] and the Pew Research Center[4]. The dataset derived from Statista includes various elements such as a screenshot of the chart image, the accompanying data table, the title, axis labels, and expertly crafted descriptive narratives concerning the chart content. Conversely, the datasets sourced from the Pew Research Center typically lack the provision of underlying data tables for the majority of their charts. To align with our pre-training objectives, we have selectively utilized only the Statista component of the Chart2Text dataset. The quantitative details of the Chart2Text dataset are systematically tabulated in Table II, with a total of 34,811 instances documented for analysis.

3) WikiTableText.: The WikiTableText dataset [29] consists of 13,318 descriptive sentences that are aligned with 4,962 tables extracted from Wikipedia[5]. These tables were retrieved via web scraping techniques and a subset of 5,000 tables was carefully curated to ensure that each table contained at least three rows and two columns, thereby meeting a predefined structural criterion. Quantitative characteristics of the WikiTableText dataset are meticulously cataloged in Table II, which enumerates a total of 13,318 instances for subsequent analysis.

4) FeVisQA: The FeVisQA dataset, as presented in [11], represents a pivotal asset in the nascent field of DV Question Answering. This dataset amalgamates a diverse set of rules and data sources to compile a comprehensive collection of question-and-answer pairs, integral for advancing research in this domain. It covers three principal types of questions:

- *Type 1*: This question type probes the semantic interpretation of DVs. An example is, "What is the meaning of this DV ?" which is illustrated as Question 1 in Figure 1.
- *Type 2*: Stemming from the associated task of DV recommendation, this category includes questions that assess the suitability of a DV for a given dataset. For instance, "Is this DV suitable for the given dataset?" The answers are structured to affirm compatibility or denote incompatibility, thus evaluating the alignment between a DV and its corresponding dataset.
- *Type 3*: Questions pertaining to data retrieval and the structural aspects of DV. These are generated using a rule-based approach, ensuring a robust and consistent set of questions and answers. Question 3 and Question 4 in Figure 1 serve as exemplary instances of this category.

Comprehensive statistics of the FeVisQA dataset are encapsulated in Table III. Similar to NVBench, the FeVisQA leverages the 152 databases originating from the Spider dataset [27], comprising a total of 79,305 free-form question-answer pairs.

TABLE III: The statistics of the FeVisQA dataset

	Number of instances			Number of questions		
Spilt	databases	QA pair	DV query	Type 1	Type 2	Type 3
Train	106	54406	9169	4799	9166	31272
Valid	16	9290	1603	844	1579	5264
Test	30	15609	2542	1453	2501	9113
Total	152	79305	13313	7096	13246	45650

B. Data Pre-processing

To enhance the data quality and ensure compatibility with downstream tasks, we instituted the following pre-processing. Initially, we excluded incomplete natural language question samples (34/25662) from the NVBench dataset. Subsequently, to prevent sequence truncation during the Bidirectional Dual-Corpus objective—which operates with a fixed token length—we retained only those entries in the Chart2Text dataset where the total number of cells (determined by multiplying the number of rows by the number of columns) did not exceed 150. This step was deemed unnecessary for the WikiTableText dataset, as it inherently possesses a maximum cell count of 108, as delineated in Table II. After employing the filtration and encoding methods described in Sections III-B, III-C, and III-D, we constructed our pretraining corpus based on the type of data. The corpus is bifurcated into two segments:

Dual-Corpus Objectives Datasets. This segment is arranged according to the following mappings:

- NL+ Schema $\leftrightarrow$ DV query
- DV query + Schema $\leftrightarrow$ Description
- Table $\leftrightarrow$ Description
- Question + DV query + Schema + Table $\leftrightarrow$ Answer

As shown in Figure 5, the aforementioned four data types are sequentially presented.

MLM Objectives Datasets. This segment amalgamates NL questions and database schemas from NVbench, DV queries, questions and answers from FeVisQA, and tables with their

[3]https://www.statista.com/

[4]https://www.pewresearch.org/

[5]https://www.wikipedia.org/

descriptions from Chart2Text and WikiTableText. These elements are integrated and then utilized to formulate the Masked Language Model (MLM) pretraining tasks. To illustrate this, a sample DV query from NVBench, which has been subjected to masking, is provided in Figure 5.

C. Data Partitioning

After preprocessing the data, we proceeded with the partitioning process. Originating from the Spider dataset [27], NVBench features a wide range of domains, including academic, railway, and scholar, which is conducive to cross-domain evaluation. The data from NVBench was divided into training, validation, and testing subsets, constituting 70%, 10%, and 20% of the dataset, respectively, to facilitate this cross-domain assessment. Furthermore, considering that FeVisQA utilizes databases from Spider, we maintained consistency with NVBench by applying the same cross-domain partitioning scheme. The partitioning of the data adheres to the original division as specified in Table II.

V. Experiments and Results

To comprehensively assess our pre-trained architecture and promote further study, we have assembled the Jointly Understanding Text and Data Visualization benchmark. This benchmark encompasses four extensively studied tasks: text-to-vis (Section V-B), vis-to-text (Section V-C), FeVisQA (Section V-D), and table-to-text (Section V-E). We incorporate established datasets pertinent to these tasks. For each task, we delineate the task definition, baselines, evaluation metrics, corresponding results, and case studies. Additionally, we perform ablation studies on the critical design elements.

A. Implementation Details

We conducted the pre-training of DataVisT5 over the course of five epochs using four NVIDIA 40GB A40 GPUs. And we standardized the maximum sequence lengths for both the input and output at 512 tokens. Our training regimen adopted a linear warm-up schedule with a 0.1 warm-up rate and set the learning rate to 5e-6. For optimization, we utilized the DeepSpeedCPUAdam optimizer with a weight decay of 0.01. Further enhancing our training efficiency, we implemented DeepSpeed's ZeRO Stage 2 offloading strategy with mixed precision (FP16) as described in [30].During the fine-tuning phase, the model exhibited significant sensitivity to hyperparameters, notably the learning rate and training epochs. A grid search was executed to determine the optimal parameters, with selection based on the performance metrics from the validation set across all models. Specifically, for multi-task fine-tuning, parameter optimization was informed by the mean performance across four tasks.

B. Text-to-Vis

Defination. For a natural language query $\{q, S\}$ consisting of a question q that articulates a user's request for a visualization and S, the schema of the relevant database D, the goal of the text-to-vis task is to generate the appropriate DV query y.

Baselines. We evaluate DataVisT5 against several established baselines for the text-to-vis task. The *Seq2Vis* approach [9] interprets the task as machine translation using a Seq2Seq model equipped with attention. The renowned *Transformer* architecture [31] and the *ncNet* framework [8], which enhances the Transformer with attention-forcing, serve as additional baselines. *RGVisNet* [26] utilizes a two-stage process for retrieving DV queries and modifying the prototype. For the performance of LLMs, we explored in-context learning through 5-shot similarity prompting with *GPT-4* [32] and fine-tuning open-source LLMs such as *Llama2-7b* [33] and *Mistral-7b* [34] using LoRA [35]. Using the *CodeT5+* model [18] as our base architecture, we employ single-task fine-tuning (SFT) without our novel pretraining as a comparison.

Task-specific Corpus. For the fine-tuning phase of our text-to-vis task, we engaged the NVBench dataset, which was delineated in Section IV-A1, originally derived from our pre-training datasets. Contrasting with the pre-training phase, the fine-tuning was conducted with a singular training objective: NL + Schema $\rightarrow$ DV query.

Evaluation Metrics. The performance evaluation of our experiment adopts four metrics, analogous to those utilized in [9]. Before delving into the specifics, it is necessary to know that each DV query comprises three key elements: the type of visualization (such as bar chart), the configuration of axis (x/y/z), and data with transformation functions (e.g. group). Additionally, let N denote the total count of test samples. The metrics are: (1) Exact Match (EM), which requires a complete match between the predicted and reference DV queries ($EM = N_{equal}/N$), (2) Visualization EM (Vis EM), assessing the accuracy of predicted visualization types ($Vis\ EM = N_{vis}/N$), (3) Data EM, focused on data points with transformation functions ($Data\ EM = N_{data}/N$), and (4) Axis EM, evaluating the congruence of axis components ($Axis\ EM = N_{axis}/N$).

Results. Results from Table IV show that foundational models like Seq2Vis and Transformer underperform in cross-domain settings. Compared to the previous state-of-the-art, RGVisNet, our multi-task finetuned model exhibited a significant 46.15% improvement in the EM metric on datasets without join operations. Furthermore, it outperformed the in-context learning approach using GPT-4 in scenarios involving join operations, enhancing the EM metric by 44.59% and 49.2%. Notably, in these scenarios, where models such as ncNet and RGVisNet have historically struggled, our model achieved an EM of 0.3451. In comparison to high-parameter (7b) open-source LLMs, our 220M DataVisT5 model performed comparably, while the 770M DataVisT5, with only 11% of the parameters, achieved optimal performance.

Case Study. We illustrate the effectiveness of our DataVisT5 model in generating DV queries compared to other baseline models in Table V. When processing a NL input, the Seq2Vis model fails to recognize essential keywords such as *visualize* and *group by*, and incorrectly identifies the chart type as *scatter*. The Transformer model, although correct in predicting the visualization type, omits significant information. A similar limitation is observed with ncNet, which, despite generating complex DV queries, fails to include the *group by* transforma-

TABLE IV: Comparative evaluation of text-to-vis models and LLMs performance on the cross-domain NVBench test dataset: non-join operations and complete NVBench with join operations. Best results are highlighted in bold.

Model	**Setting**	**NVBench w/o join operation**				**NVBench w/ join operation**			
-		Vis EM	Axis EM	Data EM	EM	Vis EM	Axis EM	Data EM	EM
Seq2Vis		0.8027	0.0000	0.0024	0.0000	0.8342	0.0000	0.0000	0.0000
Transformer		0.8598	0.0071	0.0646	0.0024	0.9798	0.0021	0.0404	0.0000
ncNet		0.9311	0.2442	0.5152	0.1465	—	—	—	—
RGVisNet		0.9701	0.5963	0.5423	0.4675	—	—	—	—
CodeT5+ (220M)	+SFT	0.9795	0.7889	0.6239	0.6010	0.9843	0.4065	0.3425	0.2968
CodeT5+ (770M)	+SFT	0.9827	0.7850	0.6696	0.6668	0.9865	0.4024	0.3713	0.3399
GPT-4 (5-shot)	+Similarity	0.9700	0.5507	0.6425	0.4726	0.9790	0.2755	0.3708	0.2313
LLama2-7b	+LoRA	0.9323	0.7432	0.6203	0.6420	0.9446	0.4281	0.3174	0.3327
Mistral-7b	+LoRA	0.9821	0.7753	0.6649	0.6761	0.9246	**0.4310**	0.3386	0.3374
DataVisT5 (220M)	+MFT	0.9827	**0.8078**	0.6680	0.6688	0.9873	0.4123	0.3586	0.3324
DataVisT5 (770M)	+MFT	**0.9850**	0.7983	**0.6770**	**0.6833**	**0.9884**	0.4112	**0.3863**	**0.3451**

tion function. RGVisNet accurately maps the term ’price’ to the ’baseprice’ column in the rooms table but does not produce the correct aggregate functions, avg and min. The SFT CodeT5+ incorrectly predicts the elements for *group by*. In contrast, our MFT DataVisT5 model accurately constructs the query: ”*visualize scatter select avg(rooms.baseprice), min(rooms.baseprice) from rooms group by rooms.decor*”, uniquely achieving the correct visualization results.

C. Vis-to-Text

Definition. When provided with a DV query q and a database D that includes a schema S, the vis-to-text task is focused on creating an intelligible textual description z that explains the DV query within database schema.

Baselines. For our evaluation, we selected several established models and LLMs: an enhanced *Seq2Seq* model, which incorporates an attention mechanism as described by [31] to improve its interaction between the encoder and decoder; the vanilla *Transformer* model as introduced in the context of text-to-vis tasks; *BART* [36], a transformer-based model that combines bidirectional encoding with auto-regressive decoding; *CodeT5+*, our base architecture; *GPT-4* in a zero-shot setting; and *Llama2-7b* and *Mistral-7b*, both with LoRA fine-tuning.

Task-specific Corpus. The unidirectional training target for the vis-to-text task was structured as DV query + Schema → Description. We employed the NVBench dataset, as referenced in Section IV-A1, analogous to the dataset used for the text-to-vis task. A notable distinction for the vis-to-text task lies in the inherent one-to-many relationship, where a singular DV query may correspond to multiple descriptions. To establish a definitive corpus for subsequent fine-tuning and evaluation, we selected a single representative description from the multiples.

Evaluation Metrics. To assess the quality of the generated textual descriptions, we employed three metrics: BLEU [37], ROUGE [38], and METEOR [39]. (1) BLEU measures the precision of N-gram overlaps with reference texts, modified by a brevity penalty. (2) In contrast, ROUGE emphasizes recall, assessing the extent of n-gram overlap. (3) METEOR surpasses BLEU in mimicking human judgement by considering exact matches, stemming, synonyms, and penalizing for word order differences. Specifically, we report BLEU scores for unigram, bigram, and four-gram levels (BLEU-1, BLEU-2, BLEU-4), and ROUGE F1 scores for unigrams (ROUGE-1), bigrams (ROUGE-2), and longest common subsequences (ROUGE-L).

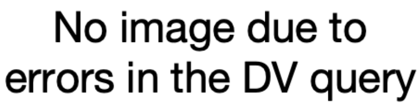
Results. As detailed in Table VI, the traditional Seq2Seq and Transformer models significantly underperform compared to other models, limited by their parameter size. Although GPT-4 outperforms traditional models in a zero-shot setting, the SFT BART, benefiting from its structure that combines context awareness with autoregressive features, shows superior performance. Moreover, LoRA fine-tuned open-source LLMs Llama2-7b and Mistral-7b, even with larger parameters, do not perform as well as BART, which has significantly fewer parameters, in the vis-to-text task. Despite our base architecture CodeT5+, enhanced through single-task fine-tuning, showing competitive performance, our proposed DataVisT5 in both 220M and 770M configurations achieves the best performance.

Case Study. In the comparative analysis presented in Table VII, the Seq2Seq model produces outputs that significantly deviate from the ground truth, indicating a disjointed understanding. The Transformer model, while capturing the basic structure of a bar chart and its ascending order, uses imprecise language that muddles the details. The SFT BART model makes progress by accurately suggesting a bar chart in ascending order but is hampered by suboptimal phrasing. The SFT CodeT5+ model, although closely aligned with the ground truth, fails to grasp the significance of the term *lname* in the visualization context. In stark contrast, our DataVisT5 model, powered by a 770M parameter architecture and enhanced through MFT, excels by providing a concise and clear directive that adeptly delineates the required bar chart with an ascending Y-axis, categorizing students by last name who are without food allergies, thus closely mirroring the ground truth.

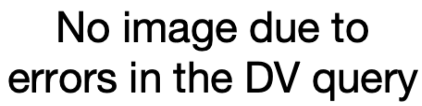
D. FeVisQA

Definition. The FeVisQA task is designed to formulate an answer A to a DV-related question Q, by leveraging a database D that encompasses a schema S and tables T, all in service of elucidating DV concepts.

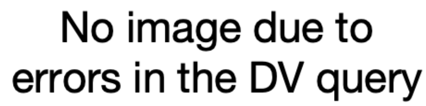
Baselines. In addressing the FeVisQA task, we adopted the same ensemble of baseline models previously applied to the vis-to-text task. This ensemble includes an attention-enhanced



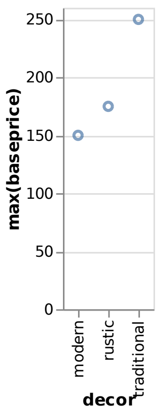
(a) Seq2Vis



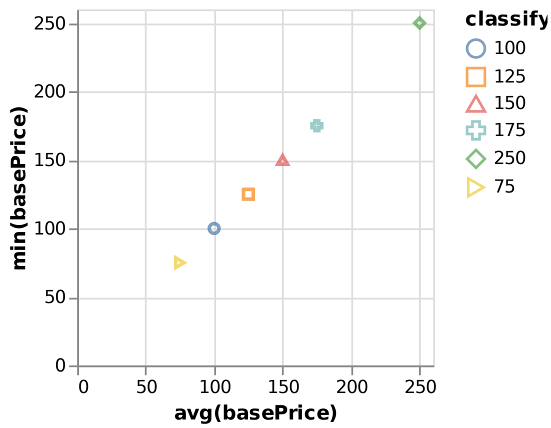
(b) Transformer



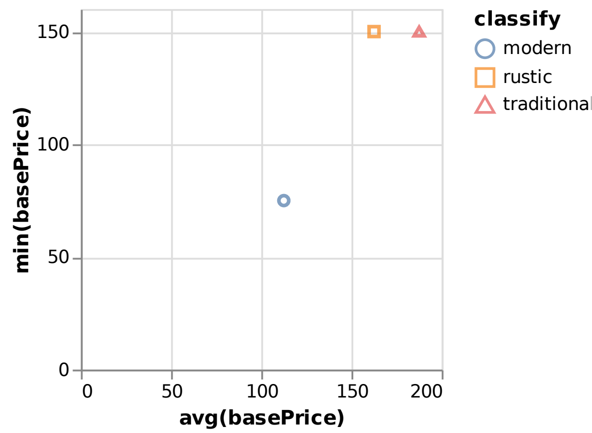
(c) ncNet



(d) RGVisNet



(e) CodeT5+



(f) Ours

Fig. 6: Visualization formats of DV Query generated by various text-to-vis models

TABLE V: The DV query examples generated by various text-to-vis models from NVBench

NL Question	Just show the average and minimum price of the rooms in different decor using a scatter.
Database Schema	| inn_1 | rooms : rooms.roomid, rooms.roomname, rooms.bedtype, rooms.baseprice, rooms.decor
Ground-truth	visualize scatter select avg(rooms.baseprice), min(rooms.baseprice) from rooms group by rooms.decor
Seq2Vis (×)	visualize bar select location, count(company.location) from company group by company.location → *Figure 6a*
Transformer (×)	visualize scatter select addresses.address_id, election.vote_percent from → *Figure 6b*
ncNet (×)	visualize scatter select rooms.name, rooms.employee_id from rooms where rooms.first_name like '%s%' → *Figure 6c*
RGVisNet (×)	visualize scatter select max(rooms.baseprice), rooms.decor from rooms → *Figure 6d*
CodeT5+ (×)	visualize scatter select avg(rooms.baseprice), min(rooms.baseprice) from rooms group by rooms.baseprice → *Figure 6e*
Ours (✓)	visualize scatter select avg(rooms.baseprice), min(rooms.baseprice) from rooms group by rooms.decor → *Figure 6f*

TABLE VI: Comparative performance analysis of models and LLMs for vis-to-text task . Best results are highlighted in bold.

Method	**Setting**	BLEU-1	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Seq2Seq		0.2766	0.1520	0.0296	0.3571	0.1343	0.2893	0.2528
Transformer		0.2825	0.1635	0.0345	0.3634	0.1476	0.2958	0.2755
BART	+SFT	0.4301	0.2892	0.1009	0.4721	0.2209	0.3647	0.4586
CodeT5+(220M)	+SFT	0.4431	0.3060	0.1236	0.4873	0.2403	0.3770	0.4872
CodeT5+(770M)	+SFT	0.4518	0.3154	0.1278	0.4898	0.2431	0.3928	0.4965
GPT-4 (0-shot)		0.3843	0.2210	0.0387	0.4180	0.1527	0.2925	0.4350
LLama2-7b	+LoRA	0.3029	0.1520	0.0314	0.3581	0.1055	0.2733	0.3028
Mistral-7b	+LoRA	0.3512	0.2431	0.0897	0.4402	0.2158	0.3549	0.3925
DataVisT5 (220M)	+MFT	0.4584	**0.3160**	0.1245	**0.5000**	0.2437	0.3978	**0.4986**
DataVisT5 (770M)	+MFT	**0.4566**	0.3155	**0.1332**	0.4974	**0.2460**	**0.3986**	0.4851

Seq2Seq model, the *Transformer* model, the SFT versions of the base *BART* and *CodeT5+* models, along with a zero-shot *GPT-4*, and LoRA fine-tuned *Llama2-7b* and *Mistral-7b*.

Task-specific Corpus. The FeVisQA task necessitated the formulation of a unidirectional training objective, structured as: Question + DV query + Schema + Table → Answer. We utilized the FeVisQA dataset for this purpose, which is elaborated upon in Section IV-A4 and originates from pre-training datasets.

Results. From Table VIII In the FeVisQA task, the MFT DataVisT5 model with 770M parameters outperforms competitors across all metrics. Compared to SFT CodeT5+ with an identical parameter setting of 770M, DataVisT5 exhibited

TABLE VII: The description examples generated by various vis-to-text methods

DV query	*Figure 7* → visualize bar select student.lname, count(student.lname) from student where stuid not in (select has_allergy.stuid from has_allergy join allergy_type on has_allergy.allergy = allergy_type.allergy where allergy_type.allergytype = 'food') group by lname order by count(student.lname) asc
Database Schema	∣ allergy_1 ∣ allergy_type : allergy_type.allergy, allergy_type.allergytype ∣ has_allergy : has_allergy. stuid, has_allergy.allergy ∣ student : student.stuid, student.lname, student.fname, student.age, student.sex, student.major, student.advisor, student.city_code
Ground-truth	List the last name of the students who do not have any food type allergy and count them in a bar chart, show Y-axis from low to high order.
Seq2Seq (×)	for a bar chart for the the number of the that have the that have the , and a bar chart, and a bar chart.
Transformer (×)	Find the last names that some last name when that are not steered by any last name as well using a bar chart , and rank by the number of last name in asc .
BART (×)	A bar chart for finding the number of the names of all students who do not have any allergy with the allergy type "Food", and could you display in ascending by the y-axis?
CodeT5+ (×)	Find the number of students who do not have any allergy type for food in each lname with a bar chart.
Ours (✓)	Give the number of students who do not have any allergy for food in each last name, show by the y-axis from low to high with a bar chart.

TABLE VIII: Comparative performance analysis for FeVisQA and table-to-text tasks highlighted by top metric scores.

Method	Setting	FeVisQA				Table-to-Text			
-		BLEU-1	ROUGE-1	ROUGE-L	METEOR	BLEU-4	ROUGE-1	ROUGE-L	METEOR
Seq2Seq		0.3642	0.3755	0.3683	0.1955	0.1575	0.4539	0.3995	0.3324
Transformer		0.2868	0.2984	0.2903	0.1556	0.0875	0.3838	0.3152	0.2642
BART	+SFT	0.7379	0.7391	0.7290	0.4376	0.3824	0.6314	0.5549	0.5845
CodeT5+(220M)	+SFT	0.6813	0.6801	0.6694	0.4086	0.3814	0.6183	0.5450	0.5844
CodeT5+(770M)	+SFT	0.7039	0.7032	0.6930	0.4211	0.3848	0.6284	0.5511	0.5946
GPT-4 (0-shot)		0.1148	0.1731	0.1599	0.2312	0.1565	0.4277	0.3281	0.4146
LLama2-7b	+LoRA	0.4214	0.4336	0.4223	0.2582	0.2010	0.4988	0.4523	0.3923
Mistral-7b	+LoRA	0.7404	0.7671	0.7574	0.4251	0.2003	0.5002	0.4538	0.3948
DataVisT5 (220M)	+MFT	0.7164	0.7158	0.7051	0.4273	0.3822	0.6259	0.5478	0.5926
DataVisT5 (770M)	+MFT	**0.7893**	**0.7895**	**0.7788**	**0.4671**	**0.4199**	**0.6520**	**0.5775**	**0.6227**

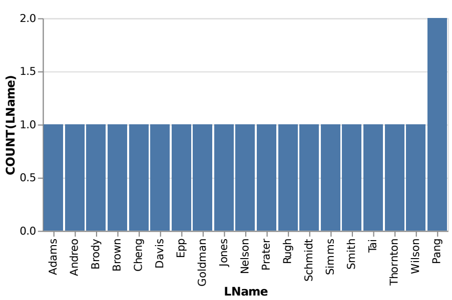


Fig. 7: Visualization Chart

a significant 10.92% increase in the METEOR score post fine-tuning, underscoring its remarkable proficiency in answering free-form questions. This enhanced performance can be attributed to the integration of textual information and DV knowledge during the DataVisT5 pre-training phase, which effectively facilitates the model's understanding of the complex cross-domain relationship between text and DV.

Case Study. Upon reviewing the outcomes documented in Table X, we observe that the Seq2Seq, Transformer, and SFT BART models exhibit various discrepancies from the ground truth. The Seq2Seq model consistently produces incorrect responses, indicating significant misalignment. The Transformer model correctly identifies the smallest chart segment but lacks consistency in other queries. SFT BART correctly identifies the number of chart segments but often overestimates numerical values. While the SFT CodeT5+ model answers most questions correctly, it inaccurately responds to *"What is the total number of count(film.type)?"*. In contrast, our DataVisT5 model is the only one that consistently provides accurate answers across both binary and numerical inquiries.

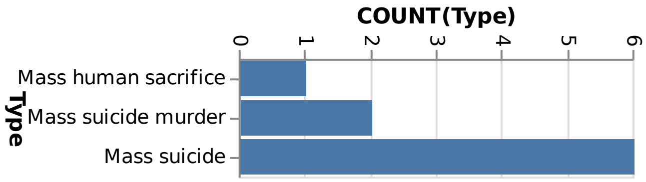
E. Table-to-Text

Defination. With a table T as the input, the table-to-text task is concentrated on producing a clear, readable narrative z that captures and clarifies the essence of the data within T.

Baselines. Consistent with the previous vis-to-text and FeVisQA tasks, which also focus on text generation modalities, we selected foundational seq-to-seq models for our analysis: the *Seq2Seq* with an attention mechanism, the original *Transformer* model, and fine-tuned versions of the base *BART* and *CodeT5+* models, specifically tailored for single-task applications. Additionally, we included a zero-shot *GPT-4* model and LoRA fine-tuned *Llama2-7b* and *Mistral-7b* in our evaluation.

Task-specific Corpus. For the table-to-text task, we formulated the unidirectional training target to Table → Description, utilizing a pre-processed pre-training corpus. We amalgamated two publicly accessible datasets, Chart2Text and WikiTableText, which are elaborated upon in Section IV-A2 and Section IV-A3.

Results. As shown in Table VIII, our MFT 770M DataVisT5 model outperforms competing approaches in the table-to-text task, achieving the highest METEOR score of 0.6227. This demonstrates DataVisT5's exceptional ability to generate textual descriptions from tabular data. Foundational models such as Seq2Vis and the Transformer struggle with understanding



(a) Visualization Chart

film.type	count(film.type)
Mass human sacrifice	1
Mass suicide	6
Mass suicide murder	2

(b) Table

Fig. 8: Visualization formats of DV Knowledge used in FeVisQA case study

TABLE IX: Sequence formats of DV Knowledge used in FeVisQA case study

DV query	*Figure 8a* → visualize bar select film.type, count(film.type) from film join film_market_estimation on film.film_id = film_market_estimation.film_id group by type order by type asc
Table	*Figure 8b* → ∣ col : film.type ∣ count(film.type) row 1 : Mass human sacrifice ∣ 1 row 2 : Mass suicide ∣ 6 row 3 : Mass suicide murder ∣ 2
Database Schema	∣ film_rank ∣ film : film.film_id, film.title, film.studio, film.director, film.gross_in_dollar ∣ film_market_estimation : film_market_estimation.estimation_id, film_market_estimation.low_estimate, film_market_estimation.high_estimate, film_market_estimation.film_id, film_market_estimation.type, film_market_estimation.market_id, film_market_estimation.year

TABLE X: The answer examples generated by various FeVisQA methods

DV Question	Ground-truth	Seq2Seq	Transformer	BART	CodeT5+	Ours
Is any equal value of y-axis in the chart?	No	Yes (×)	Yes (×)	Yes (×)	No (✓)	No (✓)
How many parts are there in the chart?	3	5 (×)	4 (×)	3 (✓)	3 (✓)	3 (✓)
What is the value of the smallest part in the chart?	1	2 (×)	1 (✓)	1 (✓)	1 (✓)	1 (✓)
What is the total number of count(film.type)?	9	11 (×)	12 (×)	15 (×)	10 (×)	9 (✓)

subjtitle	subjsubtitle	year	english title	publisher	notes
so ji-sub	books	2010	so ji-sub's journey	sallim	photo-essays

Fig. 9: Table used in table-to-text case study

TABLE XI: The description examples generated by various table-to-text methods

Table	*Figure 9* → ∣ col : subjtitle ∣ subjsubtitle ∣ year ∣ english title ∣ publisher ∣ notes row 1 : so ji-sub ∣ books ∣ 2010 ∣ so ji-sub's journey ∣ sallim ∣ photo-essays
Ground-truth	Sallim was the publisher of so ji-sub's journey in 2010.
Seq2Seq (×)	the format of was was was was was was.
Transformer (×)	In movie brand played in 2010.
BART (×)	So ji-sub's journey was published by photo-essays in 2010.
CodeT5+ (✓)	So ji-sub's journey was published by sallim in 2010.
Ours (✓)	Sallim was the publisher of so ji-sub's journey in 2010.

tables, while the commonly used SFT BART model performs closely to the SFT CodeT5+ (770M) but is still outpaced by DataVisT5. Moreover, the GPT-4 and open-source LLMs also underperform compared to our model. This superior performance is attributed to DataVisT5's integration of textual information and DV knowledge during pre-training.

Case Study. As detailed in Table XI, the Seq2Seq model's output significantly diverged from the ground truth, producing redundant and irrelevant text without the needed factual content. The Transformer model inaccurately identified the subject as a movie brand rather than a publisher, missing essential details. Although SFT BART correctly identified the publication year and the work's nature, it misattributed the publisher. In contrast, while the SFT CodeT5+ model's responses were semantically close to the ground truth, our model consistently generated descriptions that precisely matched the ground truth.

F. Ablation Studies

We conduct experiments to verify the effectiveness of each critical design in the proposed DataVisT5. Specifically, we establish the MFT DataVisT5 (770M) with all designed components as the baseline. We created variants of DataVisT5 by omitting the BDC objective in the pretraining stage, removing temperature up-sampling during MFT, and evaluating without MFT in a zero-shot setting. Additionally, we compare the use of SFT and MFT, and CodeT5+ versus T5-large as the starting point. From Table XII, it is evident that removing or replacing designed components results in performance degradation across

TABLE XII: Ablation study results: average metric values per task multipled by 100

Model	Method	text-to-vis		vis-to-text		FeVisQA		table-to-text		Mean	
DataVisT5 (770M)	MFT	65.22		36.18		70.62		56.80		57.21	
	w/o BDC	64.49	-0.73↓	36.16	-0.02↓	69.26	-1.36↓	55.83	-0.97↓	56.44	-0.77↓
	w/o up-sampling	62.95	-2.27↓	36.41	+0.23↑	70.69	+0.07↑	56.34	-0.46↓	56.60	-0.61↓
	w/o MFT	62.36	-2.87↓	37.12	+0.94↑	67.35	-3.27↓	53.98	-2.82↓	54.93	-2.28↓
DataVisT5 (770M)	SFT	65.01	-0.21↓	36.50	+0.32↑	70.73	+0.11↑	55.67	-1.13↓	56.98	-0.23↓
CodeT5+ (770M)	SFT	62.79	-2.43↓	35.96	-0.22↓	63.03	-7.59↓	53.97	-2.83↓	53.94	-3.27↓
T5-large	SFT	61.34	-3.88↓	33.58	-2.60↓	61.90	-8.72↓	52.03	-4.77↓	52.21	-5.00↓

the mean performance of the four tasks, which indicates the effectiveness of the critical design components in DataVisT5.

VI. Related Work

A. Pre-training for Data Engineering Tasks

Pre-trained models have been shown to be effective for language representations and beneficial for downstream tasks by substantial work [13], [14], [40]–[45]. All the success has also driven the development of machine language pretraining, which is in special format of text such as code and sql. CodeBert [46] is a bimodal pre-trained model for natural language and programming language in a bert-like architecture, showing that pretraining can improve the performance for code-related tasks. TaBert [47], TAPAS [25] and GraPPa [48] extend pre-trained models to learn a joint representation of NL text and database tables and demonstrate the effectiveness of semantic parsing tasks. Based on pre-trained language models, Rat-SQL [49] and Proton [50] enhance text-to-SQL parsing by focusing on schema linking and alignment, whereas StruG [51] specifically targets improvements in text-table alignment.

Moreover, the development of domain-adapted pre-trained models, such as CodeT5 [15] for code understanding and generation, MolT5 [16] for molecule captioning and generation, and BioT5 [52] which integrates cross-modal data in the biological domain with chemical structures and linguistic contexts, highlights the importance of specialized training beyond a generic T5 framework. These adaptations emphasize the necessity of domain-specific fine-tuning to effectively capture the contextual nuances inherent in specialized corpora.

B. DV-related Tasks

Benefiting from the convenience of visualization, various studies related to DV, including text-to-vis, vis-to-text, free-form question answering over DV and table-to-text, have attracted considerable research interest within the community. The initial text-to-vis systems were based on predefined rules or templates [53]–[56]. Although efficient, these systems were limited in their ability to handle the linguistic variability of user queries. To overcome these limitations, researchers have turned to neural network-based methods. For example, Data2Vis [57] conceptualizes visualization generation as a sequence translation task, employing an encoder-decoder neural architecture. Similarly, RGVisNet [26] initiates the text-to-vis process by retrieving a relevant query prototype, refining it through a graph neural network (GNN) model, and then adjusting the query to fit the target scenario. Concurrently, vis-to-text has been proposed as a complementary task, with improvements in performance demonstrated through a dual training framework [10]. Song *et al.* [11] further define the task of free-form question answering over DV and introduce the FeVisQA dataset, aiming to enhance the understanding of data and its visualizations.

Moreover, learning-based approaches have demonstrated exceptional performance in visually data wrangling and analytical tasks. For instance, Liu *et al.* [58] and Obeid and Hoque [59] have successfully translated visual data into textual descriptions and automated natural language summaries for charts using transformer-based architectures, respectively. Similarly, Spreafico and Carenini [60] utilized LSTM and neural networks for summarizing time-series and chart data, while Kantharaj *et al.* [28] advanced benchmarks in chart summarization. Furthermore, Juno [61], a cross-modal entity matching framework, has been developed to contextualize information retrieved from visually rich documents and gather actionable insights, thereby addressing challenges posed by the ad-hoc and often incomplete information in such documents.

VII. Conclusion

In this study, we propose DataVisT5, a novel PLM specifically designed for DV, which enhances the integration of cross-modal information in DV knowledge and natural language associations. This model introduces a unique mechanism to capture highly relevant database schemas from natural language mentions of tables, effectively unifying and normalizing the encoding of DV knowledge, including DV queries, database schemas, and tables. Our novel hybrid pre-training objectives unravel the complex interplay between DV and textual data, fostering a deeper integration of cross-modal insights. By extending the text-centric T5 architecture to adeptly process cross-modal information, DataVisT5 addresses multiple tasks related to DV with remarkable performance. Our extensive experimental results demonstrate that DataVisT5 consistently outperforms SOTA models and even higher-parameter LLMs across a wide range of DV tasks, expanding PLM applications and pushing the boundaries of what is achievable in automated data visualization and interpretation.

Acknowledgements

We thank the reviewers for their constructive feedback. Yuanfeng Song is the corresponding author. This work is supported by the following grants: [P0036742, P0038989, P0046701, P0046703, P0040041, P0040568, P0043864, P0045948, P0046453, P0048183, P0048191, P0048566, P0048887, WEB24EG01-H].

REFERENCES

[1] M. Friendly, "A brief history of data visualization," in *Handbook of data visualization*. Springer, 2008, pp. 15–56.

[2] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 93–117, 2020.

[3] D. Stodder, "Data visualization and discovery for better business decisions," *TDWI Research*, 2013.

[4] C. N. Knaflic, *Storytelling with data: A data visualization guide for business professionals*. John Wiley & Sons, 2015.

[5] J. G. Zheng, "Data visualization for business intelligence," *Global business intelligence*, pp. 67–82, 2017.

[6] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 341–350, 2016.

[7] R. A. M. Villanueva and Z. J. Chen, "ggplot2: elegant graphics for data analysis," 2019.

[8] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 217–226, 2021.

[9] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1235–1247.

[10] Y. Song, X. Huang, X. Zhao, and R. C.-W. Wong, "Natural language generation meets data visualization: Vis-to-text and its duality with text-to-vis," in *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2023.

[11] Y. Song, J. Lu, X. Zhao, R. C.-W. Wong, and H. Zhang, "Demonstration of fevisqa: Free-form question answering over data visualization," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024.

[12] Z. Kasner, E. Garanina, O. Platek, and O. Dusek, "TabGenie: A toolkit for table-to-text generation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, D. Bollegala, R. Huang, and A. Ritter, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 444–455. [Online]. Available: https://aclanthology.org/2023.acl-demo.42

[13] J. Devlin, M.-W. Chang, K. Lee, and K. 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)*, 2019, pp. 4171–4186.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[15] Y. Wang, W. Wang, S. Joty, and S. C. Hoi, "Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation," *arXiv preprint arXiv:2109.00859*, 2021.

[16] C. Edwards, T. Lai, K. Ros, G. Honke, K. Cho, and H. Ji, "Translation between molecules and natural language," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 375–413. [Online]. Available: https://aclanthology.org/2022.emnlp-main.26

[17] T. Xie, C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C.-S. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer, and T. Yu, "Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models," *EMNLP*, 2022.

[18] Y. Wang, H. Le, A. D. Gotmare, N. D. Bui, J. Li, and S. C. Hoi, "Codet5+: Open code large language models for code understanding and generation," *arXiv preprint arXiv:2305.07922*, 2023.

[19] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran, "Effortless data exploration with zenvisage: An expressive and interactive visual analytics system," *Proceedings of the VLDB Endowment*, vol. 10, no. 4, 2016.

[20] D. Li, H. Mei, Y. Shen, S. Su, W. Zhang, J. Wang, M. Zu, and W. Chen, "Echarts: a declarative framework for rapid construction of web-based visualization," *Visual Informatics*, vol. 2, no. 2, pp. 136–146, 2018.

[21] P. Hanrahan, "Vizql: a language for query, analysis and visualization," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 2006, pp. 721–721.

[22] Y. Luo, X. Qin, N. Tang, and G. Li, "Deepeye: Towards automatic data visualization," in *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 2018, pp. 101–112.

[23] Y. Luo, X. Qin, N. Tang, G. Li, and X. Wang, "Deepeye: Creating good data visualizations by keyword search," in *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 1733–1736.

[24] R. Giaquinto, D. Zhang, B. Kleiner, Y. Li, M. Tan, P. Bhatia, R. Nallapati, and X. Ma, "Multitask pretraining with structured knowledge for text-to-SQL generation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 11 067–11 083. [Online]. Available: https://aclanthology.org/2023.acl-long.620

[25] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. M. Eisenschlos, "Tapas: Weakly supervised table parsing via pre-training," *arXiv preprint arXiv:2004.02349*, 2020.

[26] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang, "Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation," ser. KDD '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1646–1655. [Online]. Available: https://doi.org/10.1145/3534678.3539330

[27] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3911–3921.

[28] S. Kantharaj, R. T. Leong, X. Lin, A. Masry, M. Thakkar, E. Hoque, and S. Joty, "Chart-to-text: A large-scale benchmark for chart summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4005–4023. [Online]. Available: https://aclanthology.org/2022.acl-long.277

[29] J. Bao, D. Tang, N. Duan, Z. Yan, Y. Lv, M. Zhou, and T. Zhao, "Table-to-text: Describing table region with natural language," in *AAAI*, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/download/16138/16782

[30] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3505–3506. [Online]. Available: https://doi.org/10.1145/3394486.3406703

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[32] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini,

S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph, "Gpt-4 technical report," 2024.

[33] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[34] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[35] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[36] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

[37] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[38] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[39] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[40] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[41] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," *arXiv preprint arXiv:1708.00107*, 2017.

[42] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, cite arxiv:1802.05365Comment: NAACL 2018. Originally posted to openreview 27 Oct 2017. v2 updated for NAACL camera ready. [Online]. Available: http://arxiv.org/abs/1802.05365

[43] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[44] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "Ernie: Enhanced representation through knowledge integration," *arXiv preprint arXiv:1904.09223*, 2019.

[45] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020. [Online]. Available: https://openreview.net/pdf?id=r1xMH1BtvB

[46] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang *et al.*, "Codebert: A pre-trained model for programming and natural languages," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 1536–1547.

[47] P. Yin, G. Neubig, W.-t. Yih, and S. Riedel, "Tabert: Pretraining for joint understanding of textual and tabular data," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8413–8426.

[48] T. Yu, C.-S. Wu, X. V. Lin, B. Wang, Y. C. Tan, X. Yang, D. Radev, R. Socher, and C. Xiong, "Grappa: Grammar-augmented pre-training for table semantic parsing," *arXiv preprint arXiv:2009.13845*, 2020.

[49] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, "Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers," 2021. [Online]. Available: https://arxiv.org/abs/1911.04942

[50] L. Wang, B. Qin, B. Hui, B. Li, M. Yang, B. Wang, B. Li, F. Huang, L. Si, and Y. Li, "Proton: Probing schema linking information from pre-trained language models for text-to-sql parsing," 2022. [Online]. Available: https://arxiv.org/abs/2206.14017

[51] X. Deng, A. H. Awadallah, C. Meek, O. Polozov, H. Sun, and M. Richardson, "Structure-grounded pretraining for text-to-sql," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.naacl-main.105

[52] Q. Pei, W. Zhang, J. Zhu, K. Wu, K. Gao, L. Wu, Y. Xia, and R. Yan, "BioT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 1102–1123. [Online]. Available: https://aclanthology.org/2023.emnlp-main.70

[53] L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger, "Data-thirsty business analysts need soda: search over data warehouse," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 2525–2528.

[54] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl, "From keywords to semantic queries—incremental query construction on the semantic web," *Journal of Web Semantics*, vol. 7, no. 3, pp. 166–176, 2009.

[55] S. Shekarpour, E. Marx, A.-C. N. Ngomo, and S. Auer, "Sina: Semantic interpretation of user queries for question answering on interlinked data," *Journal of Web Semantics*, vol. 30, pp. 39–51, 2015.

[56] W. Zheng, H. Cheng, L. Zou, J. X. Yu, and K. Zhao, "Natural language question/answering: Let users talk with the knowledge graph," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 217–226.

[57] V. Dibia and Ç. Demiralp, "Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE computer graphics and applications*, vol. 39, no. 5, pp. 33–46, 2019.

[58] C. Liu, L. Xie, Y. Han, and X. Yuan, "Automatic caption generation for visualization charts," *ele*, vol. 100, no. 1, p. 2, 2017.

[59] J. Obeid and E. Hoque, "Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model," in *Proceedings of the 13th International Conference on Natural Language Generation*, B. Davis, Y. Graham, J. Kelleher, and Y. Sripada, Eds. Dublin, Ireland: Association for Computational Linguistics, Dec. 2020, pp. 138–147. [Online]. Available: https://aclanthology.org/2020.inlg-1.20

[60] A. Spreafico and G. Carenini, "Neural data-driven captioning of time-series line charts," in *Proceedings of the International Conference on Advanced Visual Interfaces*, ser. AVI '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3399715.3399829

[61] R. Sarkhel and A. Nandi, "Cross-modal entity matching for visually rich documents," 2024. [Online]. Available: https://arxiv.org/abs/2303.00720