# Technical Report of Paper "Towards Robustness of Text-to-Visualization Translation against Lexical and Phrasal Variability"

Jinwei Lu* Yuanfeng Song†, Haodi Zhang*§ Chen Zhang¶, Kaishun WU§, Raymond Chi-Wing Wong‡
*Shenzhen University, Shenzhen, China †AI Group, WeBank Co., Ltd, Shenzhen, China
‡HKUST, Hong Kong, China §HKUST(GZ), Guangzhou, China ¶PolyU, Hong Kong, China

## I. Analysis of GRED with non-RAG Method

To compare GRED with non-RAG methods, we designed a new set of experiments. We applied data augmentation to the Fine-tuned Llama introduced in the main text. The data augmentation method involved introducing noise into the NLQ in the training set, such as adding spelling errors or random deletions, to enhance the model's robustness in schema linking. The results, shown in Table I, indicate that on the dual-variant test set of nvBench-Rob, the accuracy of the Fine-tuned Llama with data augmentation (Fine-tuned Llama w. AUG) improved by 5.25% compared to the Fine-tuned Llama without data augmentation (Fine-tuned Llama w/o. AUG). However, it still lagged behind GRED by 8.23%. This not only validates the potential of data augmentation in improving model robustness but also reaffirms the superiority of GRED, which does not require any data augmentation or additional model training.

| | **nvBench-Rob**$_{(nlq,schema)}$ | | | |
|---|---|---|---|---|
| **Model** | **Vis Acc.** | **Data Acc.** | **Axis Acc.** | **Acc.** |
| Seq2Vis | 94.16% | 7.45% | 7.11% | 5.50% |
| Transformer | 92.13% | 22.59% | 18.87% | 12.77% |
| RGVisNet | 96.76% | 47.04% | 34.07% | 24.81% |
| Few-Shot LLM | 97.38% | 30.37% | 30.03% | 17.18% |
| Fine-tuned Llama w/o AUG | 98.22% | 50.51% | 53.05% | 41.37% |
| Fine-tuned Llama w. AUG | **98.39%** | 56.26% | 59.56% | 46.62% |
| GRED | 98.14% | **58.48%** | **81.52%** | **54.85%** |

TABLE I: Results in **nvBench-Rob**$_{(nlq,schema)}$

## II. Detailed evaluation of each method's performance under specific metrics

To conduct a detailed analysis of how variability affects data visualization performance, we will categorize all baseline models into two groups: model training-based methods and prompting LLM-based methods. We will use Fine-tuned Llama and GRED as representatives of these two categories of baseline models.

As shown in Table II and III, we can observe that in terms of the Vis Acc. metric, neither Fine-tuned Llama nor GRED is significantly affected by variability, with fluctuations within a very small range (±1%). However, there is a substantial decline in the Axis Acc. and Data Acc. metrics. Specifically, Fine-tuned Llama, which represents the model training method, shows a **decrease of over 40%** in accuracy for both metrics. This indicates that the model has a significant deviation in understanding the XY-axis data intended to be displayed in the NLQ and also misinterprets data operations such as "`GROUP`", "`BIN`", "`ORDER`", and table "`JOIN`" when faced with variability.

GRED is also affected by variability, with a **13% decrease** in accuracy for the Data Acc. metric and a **10% decrease** for the Axis Acc. metric. However, this is much better compared to the over 40% decrease in accuracy observed with Fine-tuned Llama. Additionally, GRED outperforms Fine-tuned Llama by **13%** in overall accuracy when dealing with variability. In summary, our detailed analysis fully demonstrates the superiority of GRED in handling variability.

| | Fine-tuned Llama | | | |
|---|---|---|---|---|
| **Dataset** | **Vis Acc.** | **Data Acc.** | **Axis Acc.** | **Acc.** |
| **nvBench**$_{dev}$ | 98.66% | 91.54% | 95.26% | 91.62% |
| **nvBench-Rob**$_{(nlq,schema)}$ | 98.22% | 50.51% | 53.05% | 41.37% |

TABLE II: Performance of Fine-tuned Llama

| | GRED | | | |
|---|---|---|---|---|
| **Dataset** | **Vis Acc.** | **Data Acc.** | **Axis Acc.** | **Acc.** |
| **nvBench**$_{dev}$ | 97.29% | 71.57% | 91.20% | 68.70% |
| **nvBench-Rob**$_{(nlq,schema)}$ | 98.14% | 58.48% | 81.52% | 54.85% |

TABLE III: Performance of GRED

## III. Error Analysis that GRED Fails

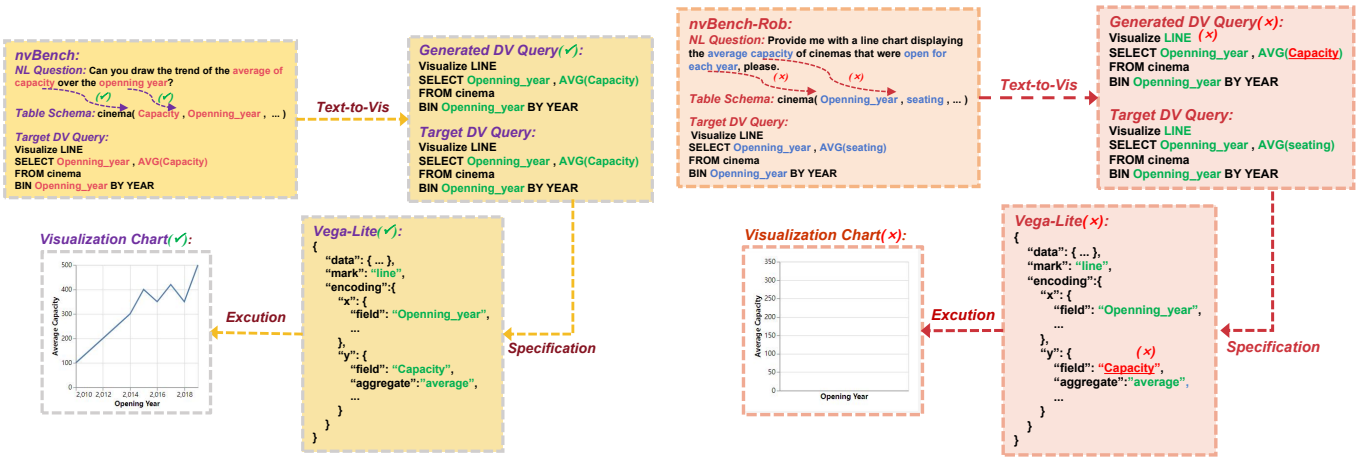
We will conduct a detailed analysis of GRED's error examples here. First, we categorize GRED's error examples into five types and select the most representative examples for display, as shown in Table IV. Below is our detailed analysis of each type of error.

- **Chart Type Error**: The main cause of errors in chart type selection is when the NLQ does not explicitly specify the type of chart to be displayed. In such cases, the model needs to infer the required chart type based on the overall semantics of the NLQ. For example, in the sample shown in Table IV, the NLQ requires displaying a ratio, which is most often represented by a pie chart. This is an implicit condition. If the model fails to analyze this, it will result in an error in chart type selection.

| | |
|---|---|
| **Chart Type Error** | |
| **NLQ** | I would like to determine the **ratio** of the number of counties for each police force. |
| **Target DVQ** | Visualize **PIE** SELECT Police_force , COUNT(*) FROM county_public_safety GROUP BY Police_force |
| **GRED** | Visualize **BAR** SELECT Police_force , COUNT(*) FROM county_public_safety GROUP BY Police_force |
| **Axis Error** | |
| **NLQ** | What is the combined **salary** and **department identification** for each department that has a workforce exceeding two employees? Please present this information in a scatter chart. |
| **Target DVQ** | Visualize SCATTER SELECT **Dept_ID** , **SUM(wage)** FROM employees GROUP BY Dept_ID |
| **GRED** | Visualize SCATTER SELECT **SUM(wage)** , **Dept_ID** FROM employees GROUP BY Dept_ID |
| **Order Error** | |
| **NLQ** | Display the mean age of drivers from the identical city of residence using a bar chart, and kindly arrange the **X-axis in descending order** from highest to lowest. |
| **Target DVQ** | Visualize BAR SELECT hometown_city , AVG(years) FROM driver GROUP BY hometown_city **ORDER BY hometown_city DESC** |
| **GRED** | Visualize BAR SELECT hometown_city , AVG(years) FROM driver GROUP BY hometown_city **ORDER BY AVG(years) DESC** |
| **Confusion between GROUP and BIN** | |
| **NLQ** | Display the variation in the total sum of Employee_ID over different Start_from bins, categorized by time, in a line chart. Please sort the Start_from bins in ascending order. |
| **Target DVQ** | Visualize LINE SELECT Start_from , SUM(EmployeeID) FROM hiring ORDER BY Start_from ASC **BIN Start_from BY YEAR** |
| **GRED** | Visualize LINE SELECT Start_from , SUM(EmployeeID) FROM hiring **GROUP BY Start_from** ORDER BY Start_from ASC **BIN Start_from BY YEAR** |
| **Misleading by RAG Examples** | |
| **NLQ** | Create a line chart illustrating the variation in the **total count of Employee_ID across different time intervals** of Start_from, and arrange the display in ascending order based on the x-axis. |
| **Target DVQ** | Visualize LINE SELECT Start_from , **SUM(EmployeeID)** FROM hiring ORDER BY Start_from ASC **BIN Start_from BY YEAR** |
| **GRED** | Visualize LINE SELECT Start_from , **COUNT(Start_from)** FROM hiring ORDER BY Start_from ASC **...** |

TABLE IV: Representative error examples of GRED on **nvBench-Rob**$_{(nlq,schema)}$ (Errors are marked with **red** colors).

- **Axis Error**: The main cause of errors in axis selection is that the NLQs in nvBench often do not explicitly specify which column should be used as the X-axis and which as the Y-axis. The model needs to analyze this based on the characteristics of the data. For example, columns using aggregate functions are more likely to be displayed on the Y-axis. In the sample shown in Table IV, the model incorrectly used "`SUM(wage)`" as the X-axis, i.e., the first column in the SELECT statement, leading to a DVQ error.
- **Order Error**: The main cause of sorting errors is that the model does not correctly understand what the data on the X and Y axes in the DVQ represent. As shown in the example in the table, the NLQ explicitly states that the data should be sorted in ascending order based on the X-axis. However, GRED still sorts the data based on the Y-axis (the second column in the SELECT statement). This error is due to the model's incorrect understanding of the DVQ's meaning.
- **Confusion between GROUP and BIN**: BIN is a keyword specific to DVQ and represents out-of-domain knowledge for LLMs. LLMs often treat DVQ as SQL, leading to confusion between GROUP and BIN. BIN means grouping a column by time, rather than grouping identical data in a column as GROUP does. In the example shown in the table, the data is grouped multiple times: first by identical "Start from" values, and then by year. This operation can result in data loss, leading to incorrect chart generation.
- **Misleading by RAG Examples**: Sometimes, the majority of retrieved examples may significantly differ from the target DVQ, which is unavoidable. When this happens,




(a) Correct Case of existing Text-to-Vis Models in the original Text-to-Vis testing set

(b) Failure Case of existing Text-to-Vis Models under robustness scenarios.

Fig. 1: Robustness Analysis Cases. These two cases demonstrate the performance of a text-to-visualization model trained on a benchmark dataset, both on the original benchmark test set and on nvBench-Rob. (a) This is a case of RGVisNet, the best-performing text-to-vis model on nvBench-Rob to date, which can effectively perform schema linking based on the NLQ and data schemas, ultimately generating the correct visualization chart. (b) This is the same case of RGVisNet on nvBench-Rob, where it needs to generate the same visualization result as in (a) (the target chart in case (b) only differs from the chart in case (a) in the axis titles). It can be observed that when the words in the NLQ no longer explicitly align with the table schemas, RGVisNet is no longer able to produce the correct visualization chart.

it can lead to situations like the one shown in the table. Not only is the aggregate function inconsistent with the target DVQ, but the operation of grouping by time is also missing, even though these conditions are explicitly mentioned in the NLQ. Upon examining the retrieved examples, it is found that the generated DVQ is very similar to the retrieved examples. This is a case where the retrieved examples mislead the model.

## IV. More Robustness Analysis Cases

Figures 1a and Figures 1b show examples of previous text-to-vis models successfully generating accurate data visualizations on the original nvBench test set, as well as instances where they fail to produce the final data visualizations due to the addition of NLQ variants and data schema variants. It is not difficult to observe that when the explicit alignment between NLQ and data schema is eliminated, previous text-to-vis models are unable to correctly perform schema linking, even when the data schema has the same meaning as the data schemas in the original training set.