

SIGN2VIS: Automated Data Visualization from Sign Language

Yao Wan^{1*} Yang Wu^{1*} Zhen Li^{1*} Guobiao Zhang^{1*} Hongyu Zhang²

Zhou Zhao³ Hai Jin^{1*} April Wang⁴

¹Huazhong University of Science and Technology, ²Chongqing University,

³Zhejiang University, ⁴ETH Zürich

wanyao@hust.edu.cn, april.wang@inf.ethz.ch

Abstract

Data visualizations, such as bar charts and histograms, are essential for analyzing and exploring data, enabling the effective communication of insights. While existing methods have been proposed to translate natural language descriptions into visualization queries, they focus solely on spoken languages, overlooking sign languages, which comprise about 200 variants used by 70 million Deaf and Hard-of-Hearing (DHH) individuals. To fill this gap, this paper proposes SIGN2VIS, a sign language interface that enables the DHH community to engage more fully with data analysis. We first construct a paired dataset that includes sign language pose videos and their corresponding visualization queries. Using this dataset, we evaluate a variety of models, including both pipeline-based and end-to-end approaches. Extensive experiments, along with a user study involving 15 participants, demonstrate the effectiveness of SIGN2VIS. Finally, we share key insights from our evaluation and highlight the need for more accessible and user-centered tools to support the DHH community in interactive data analytics.¹

1 Introduction

Data visualizations, such as bar charts, scatter plots, and histograms, effectively represent, analyze, and explore data while facilitating the discovery and communication of insights (Marriott et al., 2021). Despite the availability of numerous tools (e.g., Tableau’s Ask Data (Setlur et al., 2019) and Amazon’s QuickSight (qui, 2024)) and programming languages (e.g., Vega-Lite (Satyanarayan et al., 2016) and ggplot2 (Villanueva and Chen, 2019)),

* Also with National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Wuhan, China.

¹The dataset and source code are hosted at the project homepage: <https://sign2vis.github.io/>.

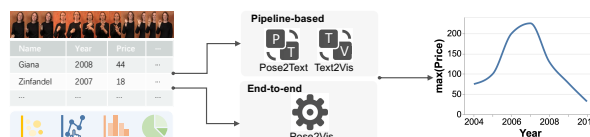


Figure 1: An overview of SIGN2VIS, with two types of approaches: pipeline-based and end-to-end.

creating effective visualizations remains challenging, especially for users with little or no prior experience. This highlights a growing need for intuitive visualization interfaces that lower the barrier for broader audiences to engage with data.

To make data analysis more accessible, growing interest has emerged in automating visualization generation from natural language—a task known as TEXT2VIS (Luo et al., 2021b,a). Existing TEXT2VIS approaches typically rely on rule-based methods (Gao et al., 2015; Setlur et al., 2016; Hoque et al., 2017) or deep learning (Luo et al., 2018, 2021b,a,c). However, these advancements largely overlook the needs of the 70 million Deaf and Hard-of-Hearing (DHH) individuals² who use over 200 distinct sign languages.

One might query: *Why not encourage the DHH individuals to submit their queries in textual format?* This overlooks a critical issue: *language deprivation*. Research (Johnson et al., 1989; Lidell, 2003) has shown that many DHH individuals primarily rely on sign language for daily communication and often exhibit limited proficiency in written English. This is largely due to the lack of auditory exposure to spoken English during crucial language-acquisition years in childhood (Holt, 1993). For example, studies indicate that many DHH high school graduates in the U.S. read at a fourth-grade level, equivalent to that of a 10-year-old (Holt, 1993). As a result, while text-based systems may work for hearing users or those flu-

²Source: The World Federation of the Deaf (<https://wfdeaf.org>).

ent in written English, we argue that supporting queries in native sign language input is essential for ensuring equitable access to data visualization for the DHH community.

The Problem: SIGN2VIS. To address this gap, this paper introduces SIGN2VIS, a sign language interface that enables DHH individuals to create data visualizations. SIGN2VIS automatically generates visual queries from sign language specifications over tabular data, complementing existing interaction modalities like mouse, keyboard, and natural language (Huenerfauth and Hanson, 2009; Frishberg et al., 1993; Shah et al., 2020).

Figure 1 provides an overview of the task of SIGN2VIS. One possible approach is to first translate sign language into spoken or written language, and then applying existing methods for TEXT2VIS (Luo et al., 2021b,a). However, this indirect pipeline-based approach has several limitations: it increases the risk of error propagation, requires additional training datasets, and incurs higher computational and synthesis costs. Alternatively, SIGN2VIS can directly translate sign language descriptions into visualization queries in an end-to-end manner, avoiding these drawbacks.

Our Work. The development of the SIGN2VIS system presents two challenges: the absence of a labeled dataset linking sign language to visualization queries, and the inherent complexity of interpreting sign languages. To address these, we create a paired dataset of sign language expressions (pose videos) and corresponding visualization queries. We also design SIGN2VISNET, a Transformer-based (Vaswani et al., 2017) neural network that encodes the semantics of sign language inputs, table schemas, and chart templates. Our technical evaluation shows that the system achieves 76.08% execution accuracy and is 4.7 \times more efficient than pipeline-based approaches.

We also conduct a user study with 15 DHH individuals to evaluate the effectiveness of SIGN2VIS for data visualization tasks. Results show that 12 participants found sign language input to be more natural and user-friendly than GUI- or text-based tools. Post-study discussions revealed additional challenges and design opportunities that could inform future research.

In summary, the key contributions of this paper are: (1) We are the first to identify and formulate the novel problem of generating visualization queries directly from sign language, aim-

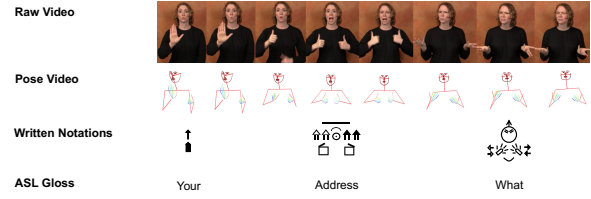


Figure 2: An example to illustrate different modalities of American Sign Language (ASL), including the video stream, pose stream, written notations, and glosses. The English translation is “What is your address?”.

ing to inspire further research on tools for improved human-computer interaction, especially for individuals with hearing disabilities. (2) We introduce a new paired dataset of sign language and visualization queries to support model exploration and training within relevant research communities. (3) We propose and benchmark SIGN2VIS with a novel Transformer-based architecture (SIGN2VISNET), and compare its performance against several pipeline-based baselines. (4) A user study shows that 12 out of 15 participants found SIGN2VIS to offer a more natural and user-friendly interface for data visualization. Additionally, we provide a demonstration of our work, which can be found in Appendix A.

2 Background

2.1 Sign Language

Sign languages differ from spoken languages as they are represented in various modalities, including videos, poses, written notations, and glosses (Figure 2). We discuss each representation below.

Raw Video Stream. Sign language communication relies on visual-gestural modalities, making video recording a natural way to capture content. However, videos often include unnecessary information for modeling, posing challenges in storage, transmission, and analysis. A lower-dimensional representation is thus preferred.

Pose Video Stream. Poses simplify video content into skeleton-like wireframes or meshes representing joint positions, offering a lower-dimensional alternative. These can be obtained through motion capture or pose estimation algorithms applied to video inputs. While motion capture provides high-quality results, pose estimation from videos is more common and less intrusive.

Written Notations. Sign language can be abstracted into discrete visual features using notation

systems. Universal systems like *SignWriting* (Sutton, 2022) and *HamNoSys* (Prillwitz and Zienert, 1990) exist alongside language-specific ones, such as *Stokoe* and *si5s* for American Sign Language and *SWL* for Swedish Sign Language. However, no single system is universally adopted.

Glosses. Glosses transcribe signs into sequences of natural-language words, preserving semantics but omitting simultaneous cues like body posture, eye gaze, or spatial relations.

Why Pose? In this paper, we formulate SIGN2VIS as translating pose videos into visualization queries. The task of sign language recognition, which converts raw videos into pose videos, is deferred to the computer vision community, where techniques for this problem have reached a mature stage (Cao et al., 2017).

2.2 Visualization Query Language

In data visualization, Vega-Lite (Satyanarayan et al., 2016) is a widely used grammar that uses a concise, declarative JSON syntax to create expressive visualizations for data analysis and presentation. However, training sequence-to-sequence models to generate hierarchical outputs like JSON Vega-Lite specifications is challenging, whereas generating sequential outputs is more feasible. To address this, Luo et al. (2021b) introduced Vega-Zero, a simplified grammar designed for TEXT2VIS. Vega-Zero converts Vega-Lite’s hierarchical structure into a sequence-based format, making it compatible with sequence-to-sequence models. Formally, a unit specification in Vega-Zero is represented as a four-tuple: (mark, data, encoding, transform). Each component has a distinct role: “mark” specifies the chart type (e.g., *bar*, *line*, *point*, or *arc* for a pie chart), “data” defines the source table, “encoding” maps columns to visual properties (e.g., x/y-axis, aggregate functions, or color), and “transform” applies data transformations like *filter*, *bin*, *group*, *sort*, or *top-k*.

2.3 Chart Templates

To reduce the scope of potential search results, Luo et al. (2021c) provided a curated collection of chart templates. Each chart template defines parameters such as the chart type (e.g., *Bar* or *Line*), the x/y-axis settings, and optional order parameters (e.g., *Descending* or *Ascending*). It is noteworthy that the introduction of these options will not contribute to the user’s operational complexity, as they can be

set as defaults. Figure 9 shows a chart template and a Vega-Zero template as examples.

3 SIGN2VIS : Task Formulation

The SIGN2VIS is formulated as a task of sequence-to-sequence learning, where the source sequence is the sign-language pose stream and the target sequence is the VQL query. Suppose that we have a sign-language corpus of N instances, i.e., $\mathcal{D} = \{(V_1, Q_1, S_1), (V_2, Q_2, S_2), \dots, (V_N, Q_N, S_N)\}$, where each instance is a triplet of sign-language pose video, VQL query, and table schema. We denote each sign-language video V as a sequence of frames $\{f_1, f_2, \dots, f_L\}$, and denote the corresponding VQL query Q as a sequence of tokens $\{q_1, q_2, \dots, q_U\}$. We denote the table schema S as a collection of column names $\{c_1, c_2, \dots, c_M\}$. The SIGN2VIS problem can be formulated as follows: given a pair of sign-language pose video V , as well the table schema S , the goal is to learn a model f that can map the input $\{V, S\}$ into a VQL query Q , i.e., $Q = f(V, S)$.

4 SIGN2VIS: The Dataset

To the best of our knowledge, the study of translation between sign language and visualization queries remains unexplored, and currently, no paired dataset of signs and visualization queries is available. Therefore, it is imperative for us to construct a SIGN2VIS dataset to facilitate further research in this domain.

4.1 From TEXT2VIS to SIGN2VIS

Creating a paired dataset of signs and visualization queries is a challenging task. One potential solution involves inviting experts in both sign language and data visualization to annotate the dataset from the ground up. Nevertheless, this approach is labor-intensive and comes with substantial human resource expenses. Instead, we translate spoken-language texts into visualization queries. A commonly used dataset for TEXT2VIS is nvBench (Luo et al., 2021a), containing 25,750 pairs of spoken-language texts and visualization queries in Vega-Zero. Inspired by TEXT2POSE techniques from computer vision, we leverage the public nvBench dataset to create a new SIGN2VIS dataset by using a pre-trained model to convert texts into pose videos (Moryossef, 2024).

Text Refinement. Before translating spoken-language text into pose videos, we analyze the text

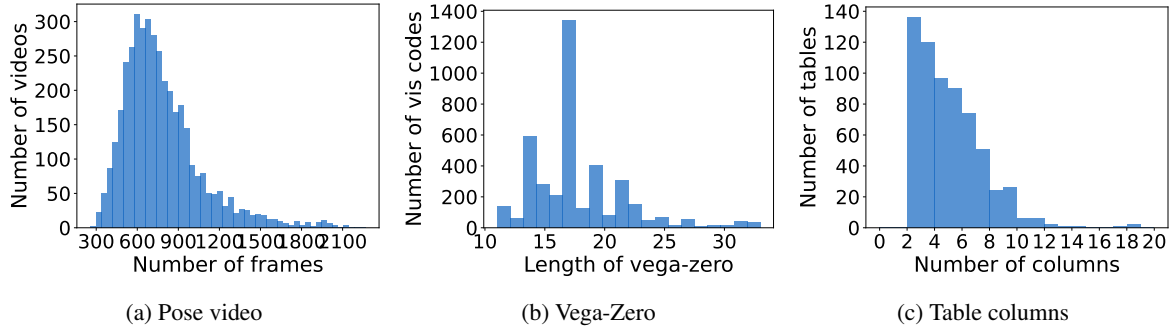


Figure 3: Overview of the distribution of SIGN2VIS.

and identify two key issues: (1) *Rare words*: Uncommon terms like “*histogram*” are often spelled out in sign language, unlike simpler alternatives such as “*bar chart*”, impacting translation quality. (2) *Verbosity*: Original sentences in nvBench are typically verbose, making them difficult to translate into sign language. To address these challenges, we use GPT-3.5 (text-davinci-003 (Brown et al., 2020)) to refine the spoken-language text with the help of the corresponding Vega-Zero template. Figure 10 illustrates this process: the green box demonstrates in-context learning with GPT-3.5, where the input query combines the spoken-language text and the Vega-Zero template. The output is a refined version of the text, which is then transformed into sign-language pose videos as described earlier.

4.2 Data Statistics

Finally, we obtain a dataset consisting of 4,007 instances comprising paired pose videos and visualization queries. In line with the nvBench configuration, our dataset is divided into three sets: 2,804 instances for training, 601 for validation, and the remaining 602 for testing. We also conduct a statistical analysis of the created SIGN2VIS dataset. Figure 3 presents an overview of the distribution of the SIGN2VIS dataset, including the distribution of (a) pose video frames, and (b) visualization queries, and (c) table columns. In Figure 3a, it is evident that a significant portion of pose video frames falls within the 500 to 900 range, emphasizing the complexity of capturing long-range dependencies among them. Turning to Figure 3b, we can find that the length of visualization queries predominantly clusters around 10. Finally, as depicted in Figure 3c, the majority of table columns fall within the 2 to 6 range.

4.3 Data Quality Assessment

We conduct a human study via an online questionnaire to evaluate the dataset we have constructed. We invite five master’s students, each with five years of software development experience, to provide expert evaluations. To ensure their proficiency, all students commit to learning American Sign Language through online resources, such as video tutorials, for at least 10 hours. For the evaluation, we randomly select 100 sign-language pose videos paired with their corresponding visualization queries. The quality of the videos is assessed on two key metrics: *consistency*, which measures alignment with the queries, and *fluency*, which evaluates the overall quality of the synthesized poses. A five-point scale is used, where 5 indicates excellence and 1 indicates poor quality. The average score is 4.3, demonstrating the high quality of our carefully curated dataset.

5 SIGN2VISNET: A Reference Approach

Figure 4 gives an overview of our proposed SIGN2VISNET, which is mainly based on the Transformer network (Vaswani et al., 2017).

5.1 Input Representation

We use frames from a sign-language pose video and the table schema as input data, along with a partially populated Vega-Zero template guided by chart templates (Figure 9c). Inspired by the *Vision Transformer* (Dosovitskiy et al., 2020), which reshapes images into sequences of flattened patches, we treat the pose video as a sequence of frames. Each frame f_i is processed through pre-trained Convolutional Neural Networks (CNNs) (Camgoz et al., 2020b), and a Transformer encoder is then applied to generate the embedding E_{f_i} . To represent the table schema, we convert it into a sequential list of column names, with each token converted into

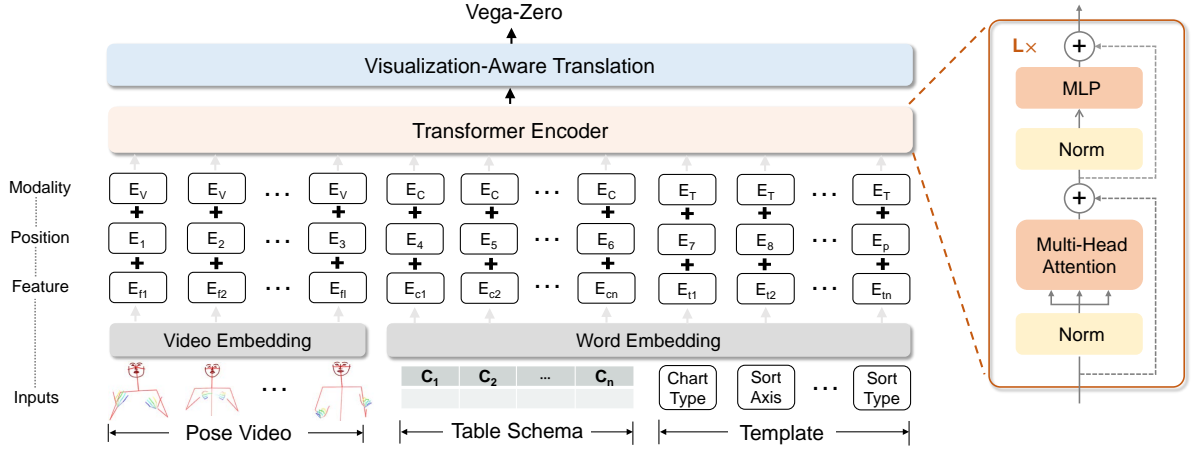


Figure 4: An illustration of the workflow of SIGN2VISNET.

a vector embedding via a word embedding layer, where the i -th column embedding is denoted as \mathbf{E}_{c_i} . Similarly, we represent the Vega-Zero template with the embedding of the i -th token as \mathbf{E}_{t_i} .

Special tokens $\langle N \rangle$ (or $\langle /N \rangle$) signify the start (or end) of video frame embeddings, while $\langle C \rangle$ (or $\langle /C \rangle$) and $\langle T \rangle$ (or $\langle /T \rangle$) mark the start (or end) of column and template embeddings. Finally, these sequences are concatenated as follows:

$$\begin{aligned} &\langle N \rangle, \mathbf{E}_{f_1}, \dots, \mathbf{E}_{f_L}, \langle /N \rangle, \langle C \rangle, \mathbf{E}_{c_1}, \dots, \mathbf{E}_{c_M}, \langle /C \rangle, \\ &\langle T \rangle, \mathbf{E}_{t_1}, \dots, \mathbf{E}_{t_N}, \langle /T \rangle. \end{aligned} \quad (1)$$

After obtaining the initial embeddings for video frames and column tokens, we enhance the representation with a modality embedding \mathbf{E}^m , indicating the source (e.g., video frames or column tokens), and a positional embedding \mathbf{E}^p , encoding their spatial arrangement in the sequence.

Using the input embedding sequence $[\mathbf{E}_1^0, \dots, \mathbf{E}_n^0]$, we process it through a Transformer encoder network (Vaswani et al., 2017), producing the following output:

$$\mathbf{O}^e = \text{TransformerEncoder}(\mathbf{E}_1^0, \dots, \mathbf{E}_n^0). \quad (2)$$

5.2 Visualization Query Generation

Following ncNet (Luo et al., 2021c), this paper employs a constrained decoder network that leverages visualization knowledge and chat templates to ensure the accuracy of the generated Vega-Zero specification. Given the final representation of the fused input pose video, the table columns, and table templates, denoted as \mathbf{O}^e , we formulate the process of the decoder network as follows:

$$\mathbf{O}^d = \text{TransformerDecoder}(\text{ctx}, \mathbf{O}^e), \quad (3)$$

where ctx denotes the embedding of previously generated tokens, and \mathbf{O}^d denotes the output of Transformer decoder (Vaswani et al., 2017).

Based on the decoder output, we pass it through a linear layer followed by softmax to estimate word probabilities across the vocabulary (Devlin et al., 2019), as follows:

$$p(q|\mathbf{O}^d) = \text{softmax}(\mathbf{W}\mathbf{O}^d + \mathbf{b}), \quad (4)$$

where \mathbf{W} and \mathbf{b} are the learnable parameters.

5.3 Model Learning and Inference

To train the SIGN2VISNET, we adopt the cross-entropy loss function, which aims to maximize the log-likelihood of predicting the next word. We use the Adam optimizer (Kingma and Ba, 2015) to iteratively update the model parameters. We integrate an enhanced beam search (Spero and Braatz, 2019) to leverage prior knowledge of visualization throughout the decoding process at each iteration.

5.4 Language-Aware Rendering

The generated Vega-Zero queries can be seamlessly translated into executable visualization programming languages, such as Vega-Lite. The adaptation of the code from Vega-Zero to Vega-Lite is based on the pioneering work of ncNet (Luo et al., 2021c), and we have made necessary modifications to align it with the current Vega-Lite version.

6 Experimental Evaluation

6.1 Baselines and Implementations

In our experiments, we carefully design several baselines for comparison, including the pipeline-based approaches and several variants of SIGN2VISNET.

	w/o Template				w/ Template			
	BLEU-4	ROUGE-L	EMR	Ex. Acc.	BLEU-4	ROUGE-L	EMR	Ex. Acc.
Pipeline-based								
SIGN2TEXT+Transformer	92.20	96.26	69.78	72.92	92.32	96.36	70.10	73.25
SIGN2TEXT+ncNet	93.28	96.67	71.93	75.08	94.06	97.15	77.08	78.24
SIGN2TEXT+GPT-3	92.01	96.45	66.61	69.44	92.60	96.78	70.60	71.93
End-to-End								
GPT-4o	20.73	52.44	0.00	0.00	39.14	66.15	0.00	0.00
Transformer	90.41	96.27	71.43	73.75	90.17	96.11	70.43	73.09
SIGNVISNET (Ours)	90.99	96.53	73.75	76.08	89.90	96.64	76.08	77.41

Table 1: The performance of different models on the testing dataset, for the task of SIGN2VIS. (Best scores are in **bold** font.)

▷ **Pipeline-Based Approaches.** We investigate three pipeline-based approaches. SIGN2TEXT+Transformer uses a Transformer encoder to embed sign-language videos, followed by a Transformer decoder to generate spoken-language text. This text is then translated into visualization queries using another Transformer model, serving as a widely adopted baseline in TEXT2VIS. Similarly, SIGN2TEXT+ncNet replaces the baseline with the advanced ncNet model (Luo et al., 2021c). Finally, SIGN2TEXT+GPT-3 employs text-davinci-003 (Brown et al., 2020) as the TEXT2VIS model, utilizing in-context learning (Dong et al., 2022) to leverage the strong generative capabilities of large language models.

▷ **End-to-End Approaches.** One of the end-to-end baselines is Transformer, a variant of SIGN2VISNET, which encodes input sign-language pose videos using a Transformer encoder and decodes them into visualization queries with a Transformer decoder. Inspired by the success of multimodal LLMs like GPT-4o (gpt, 2024), we evaluate GPT-4o’s performance in SIGN2VIS by providing tabular data, sign language video, and a prompt via an API call.

The implementations details of the baselines and our approach are referred to the Appendix B.1.

6.2 Evaluation Metrics

We follow the evaluation metrics that have been widely adopted in the evaluation of TEXT2SQL (Yu et al., 2018). We evaluate the quality of generated VQL in textual appearance by using the BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) scores, which have been widely adopted in evaluating text generation. To encourage generating VQL queries with novel syntax structure, the Execution Accuracy measures the quality of generated VQL queries based on the output of executing VQL queries. More details about

	Exe. Acc. (%)	Latency (s)	Speedup
SIGN2TEXT+ncNet	78.24	4.154	1×
SIGN2VISNET	77.41	0.877	4.7×

Table 2: The comparison of average inference latency.

the evaluation are referred to the Appendix B.2.

6.3 Performance Evaluation

Overall Performance. We present the performance of all investigated models across various evaluation metrics with and without chart templates. As shown in Table 1, the proposed SIGN2VISNET achieves results comparable to SIGN2TEXT+ncNet across all metrics and outperforms it in the key metrics of Exact Match Rate (73.75%) and Execution Accuracy (76.08%) without chart templates. Overall, visualization query generation with chart templates consistently outperforms results without templates, demonstrating their effectiveness. However, GPT-3, even with the *Vega-Zero* template, performs poorly due to frequent syntax errors, highlighting the need for improved prompt design to fully leverage large language models. Both SIGN2VISNET and SIGN2TEXT+ncNet achieve approximately 78% Execution Accuracy when paired with chart templates. Interestingly, GPT-o underperforms in SIGN2VIS, suggesting the need for further fine-tuning to adapt to this new domain. These findings underscore the notable precision and quality of visualization queries generated by our proposed approach.

Efficiency Speedup. To evaluate the efficiency of the end-to-end SIGN2VISNET, we measure the average inference latency for generating visualization queries. As shown in Table 2, SIGN2VISNET significantly outperforms the pipeline-based SIGN2TEXT+ncNet, achieving a 4.7× speedup in real-world efficiency. Importantly, this acceleration is achieved without sacrificing per-

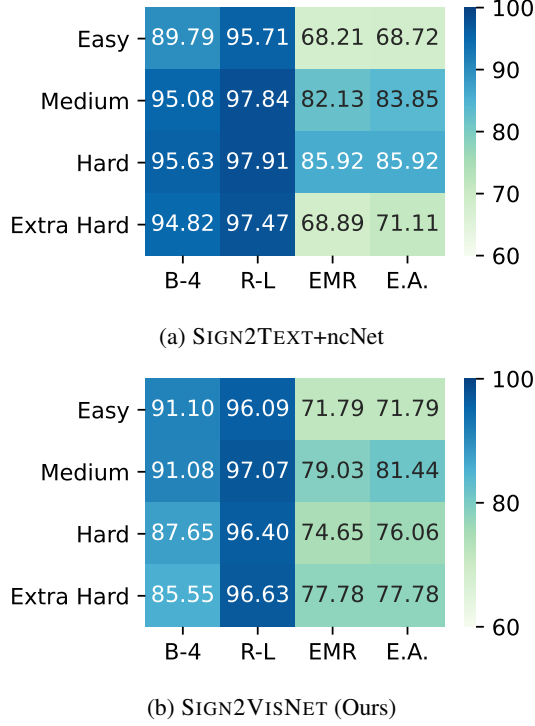


Figure 5: Comparison results when varying the complexity of visualization queries.

formance, maintaining comparable effectiveness to SIGN2TEXT+ncNet.

Performance on Various Complexity of Visualization Queries. We analyze model performance across varying visualization query complexities, predefined in the SIGN2VIS dataset. This dataset includes four complexity levels: **Easy**, **Medium**, **Hard**, and **Extra Hard**, as marked in nvBench. Figure 5 compares SIGN2VISNET and SIGN2TEXT+ncNet with chart templates under these complexities. The results show consistently high performance (BLEU-4 and Execution Accuracy) across all complexity levels. Notably, SIGN2TEXT+ncNet outperforms SIGN2VISNET at Medium and Hard levels, likely due to these levels being more sensitive to error propagation in the pipeline-based approach, where sign language is first translated into intermediate text.

Performance on Each Chart Type. To evaluate the efficacy of SIGN2VISNET, we further analyze its performance in predicting various visualization types, focusing on individual chart categories. Figure 6 details the Execution Accuracy of SIGN2VISNET on the testing dataset for four chart types: Bar, Pie, Line, and Scatter. Both SIGN2VISNET and SIGN2TEXT+ncNet show com-

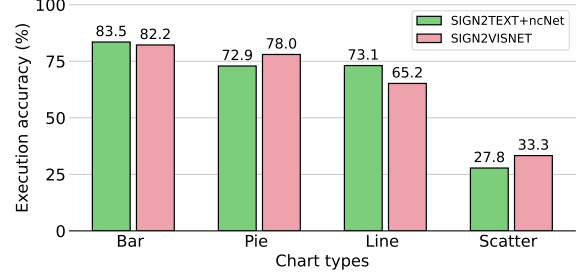


Figure 6: Execution Accuracy (%) of SIGN2VISNET on predicting each chart type of visualizations.

parable performance across the chart types, with excellent predictive accuracy for Bar, Pie, and Line charts, suggesting these are easier to predict. However, SIGN2VISNET struggles with Scatter charts due to the difficulty of predicting column names containing multiple words and underscores from pose videos, compounded by the limited representation of Scatter charts in the dataset. This issue significantly impacts the Execution Accuracy of SIGN2VISNET. To further illustrate this challenge, we present a bad case in Sec. 6.4.

6.4 Case Study and Error Analysis

To better evaluate the performance of our proposed end-to-end SIGN2VISNET and compare it with pipeline-based approaches on SIGN2VIS, we conduct a comprehensive case study and error analysis using four real-world instances from our dataset, as shown in Figure 7. For each case, given the input pose video of sign language, we present the generated Vega-Zero outputs from SIGN2VISNET and SIGN2TEXT+ncNet, alongside the ground-truth Vega-Zero. *Case A* demonstrates a successful scenario where both SIGN2VISNET and the pipeline-based SIGN2TEXT+ncNet correctly predict the visualization query, highlighting the effectiveness of both approaches.

Error Case Analysis. We also analyze two error cases to provide insights for improvement. In *Case B* and *Case C*, one approach makes the correct prediction while the other does not. In *Case B*, SIGN2TEXT+ncNet incorrectly predicts x-axis and y-axis as `city`, due to error propagation in the pipeline approach, where the word `classroom` is mistranslated as `city` during sign language translation. Conversely, in *Case C*, SIGN2VISNET mispredicts x-axis as `meter_300`, while SIGN2TEXT+ncNet is correct. This advantage arises from SIGN2TEXT+ncNet

preprocessing input during the TEXT2VIS process to focus on relevant table columns, improving accuracy when key information is translated correctly.

6.5 User Study

We also conduct a user study with 15 DHH participants to assess the effectiveness of SIGN2VIS in data visualization. Further details of the study can be found in Appendix C. The results indicate that 12 participants preferred sign language as a more natural and user-friendly alternative to GUI- and text-based tools. Additionally, the study highlighted challenges such as lack of transparency that could inform future research directions.

7 Related Work

Automated Data Visualization. Existing approaches to automated data visualization can be broadly categorized into two main groups: rule-based and deep-learning-based methods. The rule-based methods, exemplified by DataTone (Gao et al., 2015), Eviza (Setlur et al., 2016), and Evizeon (Hoque et al., 2017), rely on predefined rules and patterns to parse text and extract relevant information, which is then converted into visualizations using pre-defined mapping rules. In contrast, deep learning-based approaches, such as DeepEye (Luo et al., 2018), SEQ2VIS (Luo et al., 2021b), ncNet (Luo et al., 2021c), RGVisNet (Song et al., 2022) and HAICart (Xie et al., 2024) aim to develop neural networks, such as Recurrent Neural Networks (RNNs) (Chen and Zhuge, 2018) and Transformers (Vaswani et al., 2017), to translate natural language input into data visualizations in an end-to-end manner. Recently, various work (Madigan and Susnjak, 2023; Dibia, 2023; Cheng et al., 2023; Wu et al., 2024; Tian et al., 2024; Ouyang et al., 2025) resorts to prompting LLMs for automated data visualizations.

Sign Language Processing. Sign Language Processing (SLP) is an emerging field of artificial intelligence for the automatic processing of sign languages, requiring both NLP and computer vision techniques. Analogous to image/audio detection, identification and segmentation that are fundamental problems in computer vision, it is also important to study detection (Borg and Camilleri, 2019; Moryossef et al., 2020), identification (Gebre et al., 2013; Monteiro et al., 2016; McKee and Kennedy, 2000), and segmentation (Santemiz et al., 2009; Bull et al., 2020) for sign languages. Another in-

teresting line of work is on the sign language translation from pose estimations (Ko et al., 2019; Luong et al., 2015), glosses (Yin and Read, 2020a,b), or sign articulators from videos (Ko et al., 2019; Camgoz et al., 2020a), and sign language production (Saunders et al., 2020a,b; Zelinka and Kanis, 2020). Our work focuses on translating sign language from visualization queries, thereby serving as a pioneering interface for data visualization.

Keyboardless Programming. Keyboardless programming, which aims to provide alternative interfaces for programming, has always been a promising research direction for human-computer interaction. Typically, there are two means to achieve this goal, *i.e.*, gestured-guided programming and voice-guided programming. Programming by gestures has been widely used in controlling various IoT devices, including mobile phones (Li, 2010), robots (Waldherr et al., 2000), VR devices (Yang et al., 2019), and smart home systems (Kühnel et al., 2011). For example, Takayama et al. (2021) developed a mid-air hand gesture-based interface to manipulate spreadsheet software, such as Microsoft Excel and Google Sheets. Currently many tools, such as VoiceGrip (Desilets, 2001), VoiceCode (Désilets et al., 2006), HyperCode (Maloku and Pllana, 2016), VocalIDE (Rosenblatt et al., 2018), and Talon Voice (tal, 2023), have been developed for voiced-guided programming. In these works, the input voice is translated into commands to interact with the IDEs.

Software Accessibility. Our work also relates to software accessibility, aiming to improve access for disadvantaged communities. For example, screen readers like TalkBack on Android (Tal, 2019) and VoiceOver on iOS (Voi, 2019) assist visually impaired users in interacting with mobile devices. Chen et al. (2020) proposed a deep learning encoder-decoder network to predict labels for image-based buttons lacking descriptions. Recently, sign language has emerged as a new interface enabling DHH individuals to access information (Mahajan et al., 2022b,a; Tang et al., 2023; Mahajan, 2024; Chen et al., 2024). Cao et al. (2024) analyzed how DHH creators use sign language and other modalities on TikTok, highlighting their practices and challenges. Potluri et al. (2022) introduced CodeWalk, an extension to Microsoft’s LiveShare for supporting collaborative software engineering activities like code reviews and refactoring. Zhou et al. (2023) developed SignQuery,

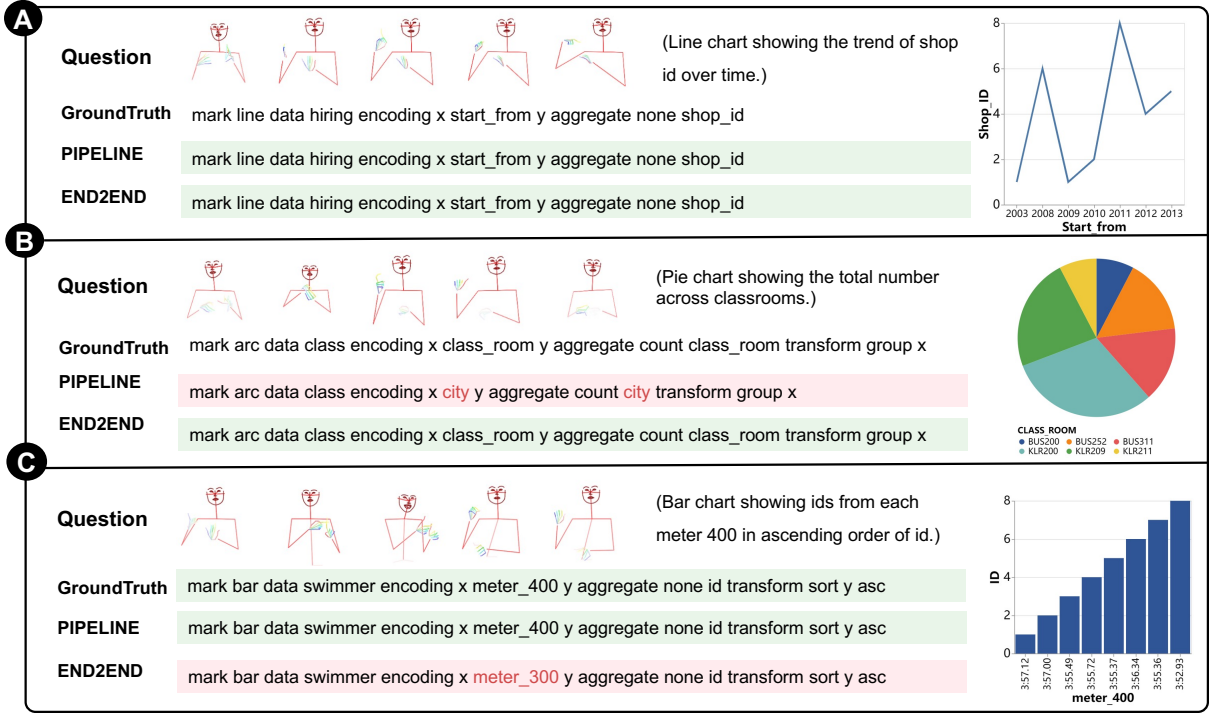


Figure 7: Case studies and error analysis.

a sign-language interface using wearable sensors to help DHH individuals perform search tasks. To our knowledge, our work is the first to design a new interface specifically for DHH individuals to perform data visualization using sign languages.

8 Discussion

In this paper, we formulate and investigate a new but important problem of SIGN2VIS. It opens a new direction to synthesize visualization queries from sign language, as a complementary interface for interacting with the database. Our work aims to design a novel human-centered interface for a large number of DHH individuals to perform data visualization. In addition to the impact on the NLP community, our work also has impacts on other related communities, including the communities of human-computer interaction, visualization, database, computer vision, and programming languages. To the communities of human-computer interaction, visualization and database, our work can inspire researchers from these communities to design better human-centered interfaces to cater to more people who are physically disabled. To the communities of computer vision, and programming languages, the created dataset and built benchmark can facilitate researchers from these communities to design better encoder networks to understand

sign language and design better generators to synthesize visualization queries with higher quality.

9 Conclusion

In this paper, we have identified a new research problem of SIGN2VIS to aid DHH individuals in data visualization, opening a path for synthesizing VQL queries from sign language as a complementary interface. We present a parallel dataset of sign language pose videos and VQL queries to support model development, along with a benchmark system featuring an end-to-end model and pipeline-based baselines for SIGN2VIS. Our work lays the groundwork for improving the accessibility of interactive data analytics for individuals with hearing impairments. We believe that our work will have a broader impact on communities of database, human-computer interaction, and programming languages. We also expect that our exploratory study on SIGN2VIS would inspire further research on this topic.

Acknowledgements

This work is partially supported by the Major Program (JD) of Hubei Province (Grant No. 2023BAA024). We would like to thank Wei Zhao and Shijie Zhang for their initial contributions to this project during their time as master students.

Limitations

One limitation of this work lies in our created dataset. In this paper, we only consider one standard of sign language, *i.e.*, American Sign Language (ASL). However, there are many other sign languages widely used in different communities or countries, such as Chinese sign language and Swedish sign language. We argue that our proposed approach can be easily extended to other sign languages as long as the paired dataset of sign language and VQL queries are created. We leave the extension of SIGN2VIS dataset to other sign languages, and the further improvement of data quality as our future work.

Another limitation lies in the pose video representation which has been commonly used to represent the sign language in this paper and prior work. We believe that the raw video representations can be transformed into an effective pose video representation, which is considered an orthogonal but interesting research direction. We invite more research from the communities of computer vision to advance this work.

References

2019. Google TalkBack source code. <https://github.com/google/talkback>.
2019. PyTorch. <https://pytorch.org>.
2019. VoiceOver. <https://cloud.google.com/translate/docs/>.
2023. Talon. <https://talonvoice.com/>.
2024. GPT-4o. <https://openai.com/index/hello-gpt-4o/>.
2024. Quicksight. <https://aws.amazon.com/quicksight>.
- Mark Borg and Kenneth P. Camilleri. 2019. Sign language detection "in the wild" with recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 1637–1641. IEEE.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Hannah Bull, Michèle Gouiffès, and Annelies Braffort. 2020. Automatic segmentation of sign language into subtitle-units. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12536 of *Lecture Notes in Computer Science*, pages 186–198. Springer.
- Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020a. Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*, pages 301–319. Springer.
- Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020b. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033.
- Jiaxun Cao, Xuening Peng, Fan Liang, and Xin Tong. 2024. Voices help correlate signs and words: Analyzing deaf and hard-of-hearing (dhh) tiktokers' content, practices, and pitfalls. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18.
- Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.
- Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhu, Guoqiang Li, and Jinshui Wang. 2020. Unblind your apps: Predicting natural-language labels for mobile gui components by deep learning. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 322–334.
- Jingqiang Chen and Hai Zhuge. 2018. Abstractive text-image summarization using multi-modal attentional hierarchical rnn. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4046–4056.
- Shi Chen, Xiaodong Wang, Weijun Li, Jingao Zhang, Yuge Qi, Jiaqi Teng, and Zhihan Zeng. 2024. Silent delivery: Practices and challenges of delivering among deaf or hard of hearing couriers. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- Liying Cheng, Xingxuan Li, and Lidong Bing. 2023. Is gpt-4 a good data analyst? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9496–9514.
- Alain Desilets. 2001. Voicegrip: a tool for programming-by-voice. *International Journal of Speech Technology*, 4(2):103–116.
- Alain Désilets, David C Fox, and Stuart Norton. 2006. Voicecode: An innovative speech interface

- for programming-by-voice. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pages 239–242.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186.
- Victor Dibia. 2023. Lida: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. *arXiv preprint arXiv:2303.02927*.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Nancy Frishberg, Serena Corazza, Linda Day, Sherman Wilcox, and Rolf Schulmeister. 1993. Sign language interfaces. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, pages 194–197.
- Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G Karahalios. 2015. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 489–500.
- Binyam Gebrekidan Gebre, Peter Wittenburg, and Tom Heskes. 2013. [Automatic sign language identification](#). In *IEEE International Conference on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013*, pages 2626–2630. IEEE.
- Judith A Holt. 1993. Stanford achievement test—8th edition: Reading comprehension subgroup results. *American Annals of the Deaf*, 138(2):172–175.
- Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. 2017. Applying pragmatics principles for interaction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):309–318.
- Matt Huenerfauth and Vicki Hanson. 2009. Sign language in the interface: access for deaf signers. *Universal Access Handbook*. NJ: Erlbaum, 38:14.
- R. E. Johnson, Scott K. Liddell, and Carol J. Erting. 1989. Unlocking the curriculum: Principles for achieving access in deaf education. working paper 89-3.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.
- Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683.
- Christine Kühnel, Tilo Westermann, Fabian Hemmert, Sven Kratz, Alexander Müller, and Sebastian Möller. 2011. I'm home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human-Computer Studies*, 69(11):693–704.
- Yang Li. 2010. Gesture search: a tool for fast mobile data access. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, pages 87–96.
- Scott K Liddell. 2003. *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press.
- C. Y. Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.
- Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. 2018. Deepeye: Creating good data visualizations by keyword search. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1733–1736.
- Yuyu Luo, Jiawei Tang, and Guoliang Li. 2021a. nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task. *arXiv preprint arXiv:2112.12926*.
- Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021b. Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1235–1247.
- Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2021c. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Paula Maddigan and Teo Susnjak. 2023. Chat2vis: Fine-tuning data visualisations using multilingual natural language text and pre-trained large language models. *arXiv preprint arXiv:2303.14292*.

- Shruti Mahajan. 2024. Towards inclusive research and resources in american signed languages. *ACM SIGACCESS Accessibility and Computing*, (137):1–1.
- Shruti Mahajan, Khulood Alkhudaidi, Rachel Boll, Jeanne Reis, and Erin Solovey. 2022a. Role of technology in increasing representation of deaf individuals in future stem workplaces. In *Proceedings of the 1st Annual Meeting of the Symposium on Human-Computer Interaction for Work*, pages 1–6.
- Shruti Mahajan, Zoey Walker, Rachel Boll, Michelle Santacreu, Ally Salvino, Michael Westfort, Jeanne Reis, and Erin Solovey. 2022b. Towards sign language-centric design of asl survey tools. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–16.
- Rinor S Maloku and Besart Xh Pllana. 2016. Hypercode: Voice aided programming. *IFAC-PapersOnLine*, 49(29):263–268.
- Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. 2021. Inclusive data visualization for people with disabilities: a call to action. *Interactions*, 28(3):47–51.
- David McKee and Graeme Kennedy. 2000. Lexical comparison of signs from american, australian, british and new zealand sign languages. *The Signs of Language Revisited: An Anthology to Honor Ursula Bellugi and Edward Klima*, pages 49–76.
- Caio D. D. Monteiro, Christy Maria Mathew, Ricardo Gutierrez-Osuna, and Frank Shipman. 2016. Detecting and identifying sign languages through visual features. In *IEEE International Symposium on Multimedia, ISM 2016, San Jose, CA, USA, December 11-13, 2016*, pages 287–290. IEEE Computer Society.
- Amit Moryossef. 2024. sign.mt: Real-time multilingual sign language translation application. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 182–186, Miami, Florida, USA. Association for Computational Linguistics.
- Amit Moryossef, Ioannis Tsochantaridis, Roei Aharoni, Sarah Ebling, and Srini Narayanan. 2020. Real-time sign language detection using human pose estimation. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12536 of *Lecture Notes in Computer Science*, pages 237–248. Springer.
- Geliang Ouyang, Jingyao Chen, Zhihe Nie, Yi Gui, Yao Wan, Hongyu Zhang, and Dongping Chen. 2025. nvagent: Automated data visualization from natural language via collaborative agent workflow. In *Proceedings of the 63th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.
- Venkatesh Potluri, Maulishree Pandey, Andrew Begel, Michael Barnett, and Scott Reitherman. 2022. Codewalk: Facilitating shared awareness in mixed-ability collaborative software development. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–16.
- Siegmund Prillwitz and Heiko Zienert. 1990. Hamburg notation system for sign language: Development of a sign writing with computer application. In *Current trends in European Sign Language Research. Proceedings of the 3rd European Congress on Sign Language Research*, pages 355–379.
- Lucas Rosenblatt, Patrick Carrington, Kotaro Hara, and Jeffrey P Bigham. 2018. Vocal programming for people with upper-body motor impairments. In *Proceedings of the 15th International Web for All Conference*, pages 1–10.
- Pinar Santemiz, Oya Aran, Murat Saraclar, and Lale Akarun. 2009. Automatic sign segmentation from continuous signing via multiple sequence alignment. In *12th IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 2001–2008. IEEE Computer Society.
- Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2016. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350.
- Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020a. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*.
- Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020b. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, pages 687–705. Springer.
- Vidya Setlur, Sarah E Battersby, Melanie Tory, Rich Gossweiler, and Angel X Chang. 2016. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 365–377.
- Vidya Setlur, Melanie Tory, and Alex Djalali. 2019. Inferring underspecified natural language utterances in visual analysis. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 40–51.
- Vraj Shah, Side Li, Arun Kumar, and Lawrence Saul. 2020. Speakql: towards speech-driven multimodal querying of structured data. In *Proceedings of the*


- 2020 ACM SIGMOD International Conference on Management of Data, pages 2363–2374.
- Yuanfeng Song, Xuefang Zhao, Raymond Chi-Wing Wong, and Di Jiang. 2022. Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22*, page 1646–1655, New York, NY, USA. Association for Computing Machinery.
- Max Spero and Jon Braatz. 2019. Improved beam search diversity for neural machine translation with k-dpp sampling.
- Valerie Sutton. 2022. *Lessons in SignWriting*. SignWriting Press.
- Yuta Takayama, Yuu Ichikawa, Buntarou Shizuki, Ikkaku Kawaguchi, and Shin Takahashi. 2021. A user-based mid-air hand gesture set for spreadsheets. In *Asian CHI Symposium 2021*, pages 122–128.
- Xinru Tang, Xiang Chang, Nuoran Chen, Yingjie Ni, RAY LC, and Xin Tong. 2023. Community-driven information accessibility: Online sign language content creation within d/deaf communities. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–24.
- Yuan Tian, Weiwei Cui, Dazhen Deng, Xinjing Yi, Yurun Yang, Haidong Zhang, and Yingcai Wu. 2024. Chartgpt: Leveraging llms to generate charts from abstract natural language. *IEEE Transactions on Visualization and Computer Graphics*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
- Randle Aaron M Villanueva and Zhuo Job Chen. 2019. ggplot2: elegant graphics for data analysis.
- Stefan Waldherr, Roseli Romero, and Sebastian Thrun. 2000. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173.
- Yang Wu, Yao Wan, Hongyu Zhang, Yulei Sui, Wucui Wei, Wei Zhao, Guandong Xu, and Hai Jin. 2024. Automated data visualization from natural language via large language models: An exploratory study. *Proceedings of the ACM on Management of Data*, 2(3):1–28.
- Yupeng Xie, Yuyu Luo, Guoliang Li, and Nan Tang. 2024. **Haichart: Human and AI paired visualization system**. *CoRR*, abs/2406.11033.
- LI Yang, Jin Huang, TIAN Feng, WANG Hong-An, and DAI Guo-Zhong. 2019. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112.
- Kayo Yin and Jesse Read. 2020a. Attention is all you sign: sign language translation with transformers. In *Sign Language Recognition, Translation and Production (SLRTP) Workshop-Extended Abstracts*, volume 4.
- Kayo Yin and Jesse Read. 2020b. Better sign language translation with stmc-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang Zifan Li James Ma, Irene Li, Qingning Yao Shanelle Roman Zilin Zhang, and Dragomir R Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.
- Jan Zelinka and Jakub Kanis. 2020. Neural sign language synthesis: Words are our glosses. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3395–3403.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.
- Hao Zhou, Taiting Lu, Kristina Mckinnie, Joseph Palagano, Kenneth Dehaan, and Mahanth Gowda. 2023. Signquery: A natural user interface and search engine for sign languages with wearable sensors. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–16.

A The SIGN2VIS System

In this section, we first describe the user interface design and implementation, followed by a usage scenario. SIGN2VIS is a cross-platform web application built with React for the front-end interface and Python for the back-end server.

A.1 User Interface Design

Figure 8 shows the Web UI of SIGN2VIS, demonstrating the scatter chart generation process. Users can start the SIGN2VIS service by clicking the “Run” button with default chart settings. Excel tables and sign-language videos are then uploaded to the cloud server. After the data upload, the model processes the selected chart properties and displays the generated visual charts on the right side of the page. The demo is available at <https://sign2vis.github.io/>.

Tabular Data Upload . We have created a UI widget that lets users upload and preview Excel spreadsheets directly on the frontend. When a user

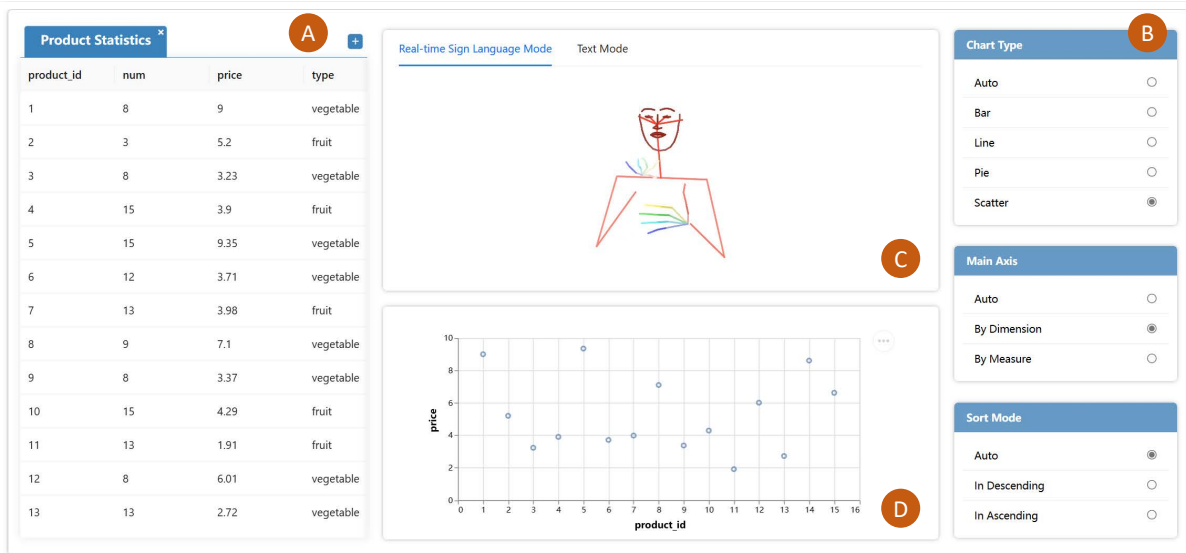


Figure 8: The Web UI of SIGN2VIS.

uploads a spreadsheet with a sign language video and clicks “Run”, the frontend converts the Excel data into JSON format.

Chart Template Setting B. To generate visual charts, we offer three adjustable attributes via drop-down menus: chart type, axis, and content ordering. Chart types include Auto, Bar, Line, Pie, and Scatter. The axis can be set to Auto, By Dimension (x-axis), or By Measure (y-axis), allowing users to control data plotting. Content ordering can be set to Auto, Descending, or Ascending, providing flexibility in organizing data.

Sign-Language Input Capturing C. We use the `navigator.mediaDevices.getUserMedia` API in HTML5 to capture real-time video from the webcam. The video stream is assigned to a `<video>` element and recorded with the `MediaRecorder` API, saving it as an MP4 Blob. After recording, the video file is stored locally as input for the SIGN2VIS system.

Visual Chart Generation D. We encode chart properties using a JSON specification. For example, to change the x-axis title, simply modify `spec.encoding.x.title = "axis title"` without altering other chart aspects. We use Vega-Lite specifications (Satyanarayan et al., 2016) due to their concise, declarative format, which maintains expressiveness. Additionally, Vega-Lite’s robust rendering engine is compatible with all major web frameworks.

A.2 User Interface Implementation

We have developed a user-friendly Graphical User Interface (GUI) that enables easy access and exploration of the results obtained from our SIGN2VIS model through a standard Web browser. The GUI is hosted on a Nginx server and employs flexible APIs provided by the Flask engine.

A.3 Usage Scenario

Automatically synthesizing visualization queries from sign language offers an intuitive interface for DHH individuals, enabling them to analyze and visualize data. This approach helps users uncover key data characteristics—such as size, time, and quantity—and identify patterns. In practice, this technology could be integrated into Tableau³, a popular platform for interactive data visualization. Currently, Tableau’s *Ask Data* feature allows users to input queries via mouse clicks or natural language, powered by NLP techniques. Extending this interface to support sign language would enable DHH users to interact with data in a language they are comfortable with, making SIGN2VIS a promising application scenario.

B More Experimental Details

B.1 Implementation Details

We implement SIGN2VISNET and all the baseline approaches using the PyTorch library (pyt, 2019), and run all experiments on a Linux server, with 128GB memory, and a single 32GB Tesla V100

³<https://www.tableau.com>

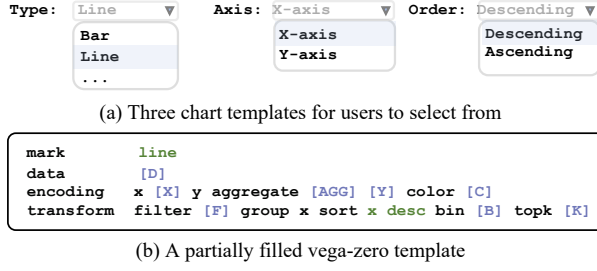


Figure 9: The chart templates and Vega-Zero template (Luo et al., 2021c).

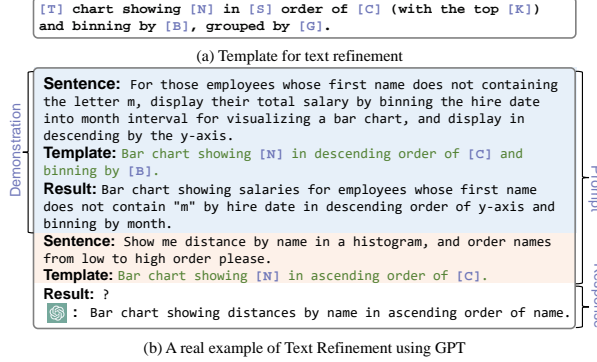


Figure 10: Natural-language queries refinement via GPT-3.5.

GPU. To encode each frame of pose videos, a 6-layer CNN is employed with one input channel and 128 output channels. For embedding the tokens of each table column and chart template, the word embedding layer is configured with a size of 256. The Transformer encoder network comprises three layers, with each layer constructed using 256 hidden units and 8 heads. Throughout the training process, a batch size of 4 is utilized. The learning rate is set to 0.0005. All the models have been validated in the validation dataset, and evaluated in the test dataset to avoid overfitting.

B.2 Evaluation Metrics

▷ **BLEU (Papineni et al., 2002)**. It is a classical evaluation metric that was first designed to evaluate the quality of translated sentences in machine translation. It measures the average n -gram precision on a set of reference sentences, with a penalty for short sentences. In this paper, we report the BLEU-4.

▷ **ROUGE (Lin, 2004)**. It is an evaluation metric based on the n -gram precision and recall. Formally, the ROUGE-N is defined as $ROUGE-N = \frac{2P_n R_n}{P_n + R_n}$, where P_n and R_n represent the n -gram precision and recall, respectively. Like ROUGE-N,

ROUGE-L is also predicated on the concept of the longest common subsequence between the generated sentence and reference sentences, as opposed to relying on n -gram statistics. In this paper, we report the ROUGE-L.

▷ **Exact Match Rate (Luo et al., 2021c)**. It quantifies the percentage of generated visualization queries that precisely align with their corresponding ground truth, serving as a metric to gauge the correctness of the generated results.

▷ **Execution Accuracy (Zhong et al., 2017)**. To encourage generating visualization queries with novel syntax structure, we employ Execution Accuracy, a metric that measures the quality of generated visualization queries based on the output of executing these queries. Even if the predicted string does not exactly match the ground truth, it is possible to get the correct visualization result.

C User Study

We report on a user study with 15 Deaf and Hard-of-Hearing (DHH) participants to understand how DHH individuals perceive the usefulness of SIGN2VIS.

C.1 Participants

We recruited 15 participants from a local DHH community using snowball sampling, starting with 3 individuals recruited via a third-party social media account linked to a regional disability organization. Eligible participants had experience with traditional GUI-based tools like Excel and were fluent in the local DHH sign language. Table 3 shows their demographic background, including age, gender, education level, and occupation.

C.2 Study Protocol

The study is conducted via an online platform we developed. Participants first answer attention check questions, then watch two videos demonstrating data visualization tasks: one using the SIGN2VIS system and the other using a baseline system, TEXT2VIS. The video order is counterbalanced to reduce bias. Afterward, participants complete a post-task survey to provide feedback. We chose recorded videos over direct tool interaction for practical reasons. Since the local DHH community uses a different sign language from standard ASL, and no dataset exists for automatic translation, we consulted a sign language expert to translate ASL recordings into the local sign language,

ID	Gender	Age	Education	Occupation
P1	M	30	3	Government Employee
P2	M	38	4	Freelance Worker
P3	F	43	4	Freelance Worker
P4	M	47	3	Freelance Worker
P5	F	47	2	Transit Staff
P6	M	47	3	Teacher
P7	M	49	2	Community Worker
P8	F	49	1	Unemployed
P9	M	51	3	Community Worker
P10	M	52	1	Unemployed
P11	M	53	1	Chef
P12	F	54	1	Retired
P13	F	54	3	Retired
P14	F	56	3	Retired
P15	M	63	1	Retired

Table 3: Summary of participants. The table shows key demographic information for 15 participants, including gender, age, education level, and occupation. Education levels are categorized as: 1 = High School/Technical, 2 = Associate Degree, 3 = Bachelor’s Degree, 4 = Graduate Degree.

simulating SIGN2VIS usage. Each session lasts 15 minutes, with participants compensated 80 CNY (about 14 dollars). The study was approved by the university’s Institutional Review Board (IRB).

C.3 Data Analysis

In the post-task questionnaire, we gathered responses to Likert-scale questions. Additionally, we collected qualitative feedback through open-ended questions. These covered participants’ perceptions of how they compare the TEXT2VIS system with the SIGN2VIS system, as well as their thoughts on GUI interfaces such as Excel. Two members of the research team collaboratively developed a coding list and conducted the thematic analysis to identify emerging themes.

C.4 Results

We describe the results from the semi-structured interview and post-task questionnaire, focusing on participants’ perceptions of the traditional GUI-based visualization tools like Excel, text-based visualization tool, and the sign language visualization tool.

C.4.1 Perceptions of Participants

We analyze participants’ perceptions when using sign language-based, text-based, and GUI-based visualization approaches.

Perceptions of the Traditional GUI-based Visualization Tool. When participants were asked to imagine using traditional GUI-based visualization tools like Excel, 12 rated it as difficult or very

difficult, with only 2 finding it very easy. Most participants reported challenges with Excel for data visualization tasks. Nine participants struggled with selecting the right formulas, while 7 found Excel’s interface difficult to navigate due to its steep learning curve. Six participants were unsure about the drag-and-drop functionality for data manipulation and visualization, and another 6 described the process as time-consuming and the UI as unfriendly.

These findings suggest that Excel, despite its capabilities, may not be well-suited for the DHH community without significant improvements in usability and accessibility.

Perceptions of the Text-Based Visualization Tool.

When evaluating the text-based visualization tool, participants showed mixed experiences. While it was somewhat more favorable than traditional GUI-based tools, challenges remain. Two participants found it "very easy" to use, but most did not, highlighting usability issues. Specifically, 10 participants struggled with the syntax and structure of text queries, making it difficult to express their desired visualizations. Additionally, 10 participants found typing queries tedious and error-prone, and 8 found the process time-consuming. Although the interface was slightly better received than Excel, 4 participants still found it lacking in user-friendliness.

These findings emphasize the need for improvements in text-based visualization tools, particularly to make them more intuitive and accessible for DHH individuals. While such tools have potential, they currently fall short in usability for this user group.

Perceptions of the Sign Language-Based Visualization Tool.

We collected feedback from DHH participants on the use of the SIGN2VIS tool compared to traditional GUI-based tools and the existing TEXT2VIS tool. Only 4 participants struggled with gestural communication, compared to more with typing. A strong majority (12 participants) found sign language a more natural and efficient way to obtain visualizations and aid data analysis. Additionally, 11 participants reported less difficulty with gesture-based input than typing, showing a clear preference for sign language as a more accessible interaction method.

C.4.2 Why Sign Language-Based Approach?

We further explore the reasons why participants prefer a sign language-based approach for data visualization.

Natural and Intuitive. The reasons participants favored sign language varied, with some highlighting its natural and intuitive nature. Participant P2 emphasized the natural and intuitive nature of sign language, noting its ability to render visual data more immediately comprehensible. The immediacy of sign language in conveying complex information was also highlighted by P8, who appreciated the convenience and speed of using sign language for automatic data visualization. P10 expressed a preference for sign language due to limitations in cultural understanding, suggesting that for some DHH individuals, sign language may serve as a more accessible and familiar medium for communication and data interpretation.

Accessibility and Flexibility. Sign language was preferred by participants such as P10 and P11 for its accessibility, particularly in daily use and as an alternative to typing. P11's reliance on sign language for daily communication underscored the practical challenges faced by those who are not adept at typing, particularly when using mobile devices. This participant's preference for sign language reflects a broader need for tools that accommodate diverse communication styles and abilities. Similarly, P14 found sign language to be more convenient, indicating a general trend among participants who value the ease of use that sign language offers in data visualization tasks.

Hybrid Approach. Interestingly, P7's suggestion to use a writing board for typing highlights a potential hybrid approach, where sign language could be complemented by written input when necessary.

C.4.3 Challenges of Using Sign Languages

We also discuss the challenges participants might encounter when using the sign language-based approach for data visualization.

AI Precision. Participants expressed mixed feelings about the accuracy of AI-generated visualizations based on sign language input. While some believed that AI could easily generate correct visualizations, the majority were either neutral or skeptical, suggesting a need for improvement in AI's ability to interpret sign language accurately. This skepticism was particularly evident among participants who had significant experience with traditional tools like Excel, who preferred manual methods due to concerns over AI precision.

Diverse Sign Languages. We also asked participants to picture what could cause errors if sign lan-

guage were used to generate visualizations. Compared to text-based visualization generation, where input errors are significantly reduced by 40%, users now face the challenge of leveraging their understanding of the data to explore and articulate more sophisticated visualizations. The cognitive process of transforming data comprehension into a visual representation demands that users not only grasp the intricacies of the dataset but also effectively conceptualize and express their visualization goals. This challenge is raised by the diversity of sign language, as noted by Participant P9, who pointed out that sign language varies regionally and culturally and is influenced by age-related factors. This diversity necessitates a tool that is adaptable and inclusive, capable of recognizing and accommodating the wide array of sign language expressions used by DHH individuals across different regions and communities.

A representative from the Chinese Federation for Disabled Persons also highlighted the challenge of promoting standard sign language across China, noting that many DHH individuals learn sign language informally. This emphasizes the importance of making the tool accessible and adaptable to diverse sign language users, aligning with the broader goals of inclusive education. In line with these challenges, the nation is actively advancing inclusive education by integrating sign language education with compulsory education, thereby enabling individuals with hearing impairments to receive education in a manner that mirrors the experiences of their hearing peers. This initiative is designed to promote social equity by providing equitable educational opportunities. Our tool aligns with these efforts by supporting individuals with hearing impairments in seamlessly transitioning between sign language learning and broader cultural education, thereby enhancing their ability to engage with and benefit from these educational processes.

C.4.4 Suggestions

The feedback from participants has been instrumental in identifying areas for improvement within the SIGN2VIS tool. A prominent suggestion was the integration of sign language input with additional complementary features, such as explanatory captions and subtitles, catering to those who might benefit from textual support alongside visual cues. Moreover, maintaining typed input as a secondary mode of interaction was highlighted as crucial, for those who are in transition from learning sign lan-

guage to incorporating it into their daily lives. In addition, participants emphasized the need for the tool to quickly and correctly interpret sign language gestures, which is essential for reducing communication barriers and enhancing the efficiency of data visualization tasks.

C.5 Limitations

The user study was constrained by the specific sign language dialect used by the local DHH community, requiring participants to observe pre-recorded queries rather than forming their own queries for actual tasks. This limitation affects the external validity of our findings, as the results may not generalize to broader populations or real-world scenarios. Future work could address this by including ASL users and incorporating the pose recognition module to allow for natural query formation.

The Prompt for Text Refinement

Sentence: for those employees whose first name does not containing the letter m, display their total salary by binning the hire date into month interval for visualizing a bar chart, and display in descending by the y-axis.

Template: Bar chart showing [N] in descending order of y-axis and binning by month.

Result: Bar chart showing salaries for employees whose first name does not contain m by hire date in descending order of y-axis and binning by month.

Sentence: [Sentence]

Template: [Template]

Result:

Figure 11: The prompt for text refinement.

The Prompt for Sign2Text+GPT-3

Sentence: "Bar chart showing the total number of wines whose price is greater than 100, grouped by year."

Template: mark bar data employees encoding x [X] y aggregate [AggFunction] [Y] color [Z] transform filter [F] group [G] sort [Y] desc topk [K] bin [B]

Result: mark bar data wine encoding x year y aggregate count year color grape transform filter price > 100 group x sort x

Table: [Table]

Sentence: [Sentence]

Template: [Template]

Result:

Figure 12: The prompt for Sign2Text+GPT-3.

The Prompt for GPT-4o

These are frames from a video that I want to upload.

1. Translate the sign language in the frames into natural language.

2. Format your answer based on the structure shown in the following examples, but do not copy the words from them. Use the format as a guide only:

- mark bar data customers_cards encoding x card_type_code y aggregate count card_type_code transform group x

- mark line data hall_of_fame encoding x yearid y aggregate count yearid transform bin x by year

- mark arc data institution encoding x type y aggregate sum enrollment transform group x

- mark point data employees encoding x commission_pct y aggregate none manager_id transform filter

first_name like '3. Return only one translated result sentence in natural language, with no additional explanations or examples.

Figure 13: The Prompt for GPT-4o