IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 20 March 2023, accepted 23 April 2023, date of publication 28 April 2023, date of current version 30 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271393

RESEARCH ARTICLE

DPVis: Automatic Visual Encoding Based on Deep Learning

JIA LIU, GANG WAN, PANKE DENG, CHU LI, SHUAI WANG, YUNXIA YIN, LEI LIU, AND YAO MU
School of Space Information, Space Engineering University, Beijing 101416, China
Corresponding author: Gang Wan (dsddfff@stu.cuz.edu.cn)

ABSTRACT Automatic visual encoding is frequently employed in automatic visualization tools to automatically map data to visual elements. This paper proposed an automatic visual encoding approach based on deep learning. This approach constructs visual encoding datasets in a more comprehensive and reliable manner to extract and label widely available visualization graphics on the Internet in accordance with three essentials of visualization. The deep learning model is then trained to create a visual encoding model with powerful generalization performance, enabling automated effective visual encoding recommendations for visual designers. The results demonstrated that our approach extends the automatic visual encoding techniques used by existing visualization tools, enhances the functionality and performance of visualization tools, uncovers previously undiscovered data and increases the coverage of data variables.

INDEX TERMS Automatic visualization, visual encoding, deep learning, visual channels.

I. INTRODUCTION

Visualization techniques utilize the human eye's capacity for perception to depict and convey data in an interactive manner, aiming at enhancing the cognition abilities of humans. The contribution of visualization techniques has enhanced people's comprehensive perception, interactive analysis, decision-making, and reasoning ability of abstract data, formed the complementary and mutual promotion of the advantages of human brain intelligence and machine intelligence, established an iterative and spiral way of information exchange and knowledge refining, and has been widely used in many fields, including scientific research, commerce, finance, medical treatment, and communication.

Visual encoding, perception, and cognition are only a few of the cognitive processes that go into data visualization [1]. The efficiency of data visualization critically depends on the accuracy of visual encoding, which is the fundamental theory and core component of data visualization. Visual encoding refers to encoding the semantics and properties of data while following certain principles and making full use of prior knowledge to reduce the time consumption of graphic perception and cognition [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

Automatic visual encoding is performed utilizing defined visual encoding rules by conventional automatic visualization techniques. These rules are principles and approaches that academics have evolved via a variety of practices. Machine learning techniques have been heavily used by automated visualization systems [3] in recent years to achieve automatic visual encoding.

This paper proposes an automatic visual encoding method based on deep learning (DPVis) that makes use of three essentials of visualization: the visual channels, scale plate, and coordinate systems, to extract visual encodings from visualization graphics and construct a visual encoding dataset for the next stage of modeling. This turns the automatic visual encoding issue into a deep learning issue and offers a viable automatic visual encoding deep learning model. This approach, which supplements the previous one, emphasized learning visual encoding specifications and building models from a wider range of available graphics, predicting visual encodings, and realizing rapid mapping of data to visual channels without resorting to predefined specifications.

The contribution of this paper is as follows:

(1) Construct visual encoding dataset. Three essentials of visualization were extracted from the graphics of the basic forms of the data types and annotated to generate a supervised visual encoding dataset.

(2) Design and implement an automatic visual encoding model. A deep learning neural network model was designed, and the visual encoding dataset was utilized to train the visual encoding model with strong generalization performance to accurately predict the visual encoding automatically.

The remainder of this paper is organized as follows. The relevant work and findings are presented in Section II. Section III describes the generation method of the visual encoding dataset in depth and the specific model and training procedure. Section IV presents the comparison experiment. The discussion of the proposed approach was presented in Section V. Finally, we conclude the contribution of DPVis along with its existing limitations and propose future work directions.

II. RELATED WORK

The demand for rapid visualization analysis has grown in recent years, and as a result, visualization has become an important part of modern data science. Consequently, there is a growing and increasingly automated requirement for visualization tools.

The growth of visualization tools may be divided into four stages [4]. First, there are tools such as D3 [5], EChart [6], VegaLite [7], and VisComposer [8] that are intended for people who are familiar with programming and visualization. In particular, ECharts is an open-sourced, web-based, cross-platform, widely used framework that facilitates the rapid createion of interactive visualization and is highly extendable, high-performance, high expandability and performance.

Second, there are tools such as iCharts [9], raw graphics [10], Polestar [11], Voyager [12], Voyager 2 [13], and iVisDesigner [14] that are made for users who are familiar with programming but not visualization. Among them, Polestar is a comprehensive set of information management and teamwork tools for intelligence analysts. Voyager seeks to complement manual chart construction with interactive navigation of a gallery of automatically-generated visualizations. Voyager 2 is a mixed-initiative system that combines manual and automated chart specification to help analysts engage in both open-ended exploration and targeted question answering. iVisDesigner is a web-based system that allows users to interactively create information visualizations for complex datasets. It achieves high levels of interactive expressiveness through conceptual modularity, which covers a wide range of information visualization design possibilities.

Third, semiautomatic visualization techniques such as SAGE [15] and BDVR [16] may produce graphics with little to no operator input.

Finally, the newest automatic visualization techniques, such as Text-to-Viz [17] and Click2Annotate [18]. Automatic visualization solutions can offer effective visual encoding recommendations without the requirement for human design and view definition for people who are unfamiliar with programming visualization. This will not lead to mistakes caused by insufficient knowledge, experience, visual illusion, cognitive dissonance, or other aspects of the manual visual encoding process.

Automatic visualization techniques aim to help people discover and explore relevant information that has been hidden in the data by automatically generating visualization graphics. The existing automatic visualization tools can be divided into three categories: rule-based, data-driven, and mixed [4].

A. RULE-BASED AUTOMATIC VISUALIZATION TECHNIQUES

Rule-based automated visualization techniques construct rules and carry out automatic visual encoding based on the Bertin-originally suggested visual channel performance ranking [19]. These rules have been validated by visualization specialists through extensive visual design experience [1].

For example, Foresight [20] method presents visualizations of the top k instances in the data based on an appropriate ranking metric to help users rapidly discover visual insights from massive, high-dimensional datasets. The MuVE [21] method is proposed for Multi-Objective View Recommendation, which may gradually and incrementally evaluate the many advantages offered by a visualization.

The Vizdeck [22] approach generates recommendations for appropriate visualizations automatically and without the need for programming. It would be better if the ranking function could connect to other data sources, but it still needs work. And Show Me [23] uses a series of user interface commands and defaults to integrate automatic visualization into commercial visual analysis system.

Alexander et al. conducted a series of experiments on cognitive biases associated with the length, font size, and height of English words [24] and revealed that using two different dimensions simultaneously in visual encoding should be avoided as much as feasible.

Szafir focused on color encoding by enhancing the color differentiation of various marker sizes and shapes to maximize the effect of visual encoding [25]. In practical applications, visual encoding must be modified in accordance with task requirements and data qualities. People arrange visual channels more precisely according to one or more assessment variables, such as perceived efficacy, user tasks, and user preferences. Szafir evaluated the effects of task and data distribution to rate the effectiveness of visual encoding, the interaction between visual channels in multidimensional visualization, and the improvement of automated visualization technologies [25].

Perry et al. offered a library of suggested graphics in accordance with the statistical features of interest. The technique offers visual keyword searches in addition to having a voting feature that allows users to change ranks, [26].

However, rule-based automatic visualization techniques are limited because they depend on a set of manually crafted interdependent rules. These rules can be complex, voluminous, and tedious update, and they may not adequately cover all edge cases to produce accurate visualizations or may not even benefit from experience and expertise in existing visual encoding.

B. DATA-DRIVEN AUTOMATIC VISUALIZATION TECHNIQUES

Data-driven automatic visualization techniques can automatically recommend visual encodings by training machine learning models to provide more effective data exploration [27], [28], [29], [30]. They do not require professional expertise or prior visualization experience, but can also quickly create data visualization. Hu et al. proposed a visual recommendation technique based on machine learning called VizML [31], which learns the most prevalent visual designs, such as visual types and axis encoding, from many datasets and their related Plotly plots, with an accuracy rate of over 85%. Cui et al. investigated Text-to-Viz [17], a cutting-edge technique for automatically creating visualization graphics from text. First, they gathered a real-world corpus sample, manually annotated these examples and then used machine learning algorithms to train the model and forecast visualization encodings. A visualization analysis application named exploroBOT [32] was created by McAuley et al. Used common browsing and exploring metaphors in social media applications as a point of reference, user-driven as the leading, and automatically generated visualization graphics and exploration guidance paths to give users accurate data representation and suitable visual encodings.

In terms of idea expression and technological approaches, Data2Vis [33] differentiates from past rule-based approaches. It develops a sequence-to-sequence model using neural machine translation technique, converts a JSON-encoded dataset to the Vega-lite visualization standard, and stresses the use of principles that may be inferred from examples while creating graphics. Lin et al. improved the automatic visualization system to better balance automatic suggestions and user intentions, to deliver a more concise and flexible creative experience through automatic design, and to preserve predictability and control through anchor suggestions [34]. DataShot [35] is an automatic system that provides better customization and data presentation, and can automatically create configuration files from table data.

Data-driven automatic visualization techniques need to extract visual features from visualization graphics. This task lacks a clear operational standard, and the outcomes directly impact the generalization performance of machine learning models [36].

C. MIXED AUTOMATIC VISUALIZATIN TECHNIQUES

Combining these two techniques usually generates better analytical outcomes. To balance the best results of these two approaches, Luo et al. automatically generated visualizations by integrating deep learning with rules [37]. Qian et al. presented a two-stage approach [38], which indexes online examples by their visual elements in the retrieval stage, and leverages recursive neural networks to help adjust the initial draft and improve its visual appearance iteratively in the adaption stage. A visual data story generating system called Calliope [39] was proposed by Shi et al. It automatically generates visual data stories from input spreadsheets through an automatic process and facilities the easy revision of the generated story based on an online story editor. Gemini is a declarative grammar and recommendation system that Kim and Heer proposed [40] for animated changes between single-view statistics visualizations. These techniques show that new theories and research questions are always evolving, and visualization may be employed as a new type of data format for artificial intelligence processing.

Despite the fact that several automatic visualization approaches have produced promising results, not all extraction and annotation strategies for visual encodings extraction are equally effective.

III. METHODOLOGY

A. OVERALL

In this study, we first gathered a large number of widely used visualization graphics of basic data, extracted, and annotated their visual encodings in accordance with the three essentials of visualization, including visual channels, X-axis attributes, Y-axis attributes, and data types, and then constructed a supervised visual encoding dataset. After designing and training a model with strong generalization performance using deep learning techniques on this dataset, users may swiftly complete automated and successful visual designs. The flowchart is shown in Fig. 1.

B. DATA SOURCE

We manually collected a sizable number of visualization graphics from the Internet as original data sources for our research to conduct. These graphics come from a wide range of websites, such as Baidu image search. Some of the gathered images are shown in Fig. 4 as we have compiled the most popular ones with good picture effects. This is step 1 of the workflows in Fig. 1.

C. CONSTRUCTION OF VISUAL ENCODING DATASET

This section describes the construction of the visual encoding dataset, including the extraction of visual encodings in accordance with three essentials of visualization and annotation of visual elements to generate a data foundation for the subsequent model establishment, which can be more objective and accurate. The scale plate, coordinate system, and visual channels are the three essentials of visualization. The visual channels are carried by the coordinate system, which typically combines the scale plate, visual channels, and coordinate system to create the final visualization graphics. This is step 2 of the workflows in Fig. 1.

1) THREE ESSENTIALS OF VISUALIZATION

a: VISUAL CHANNELS

Visual encoding is the process of transforming data into visual variables. Encoding can be understood as design or

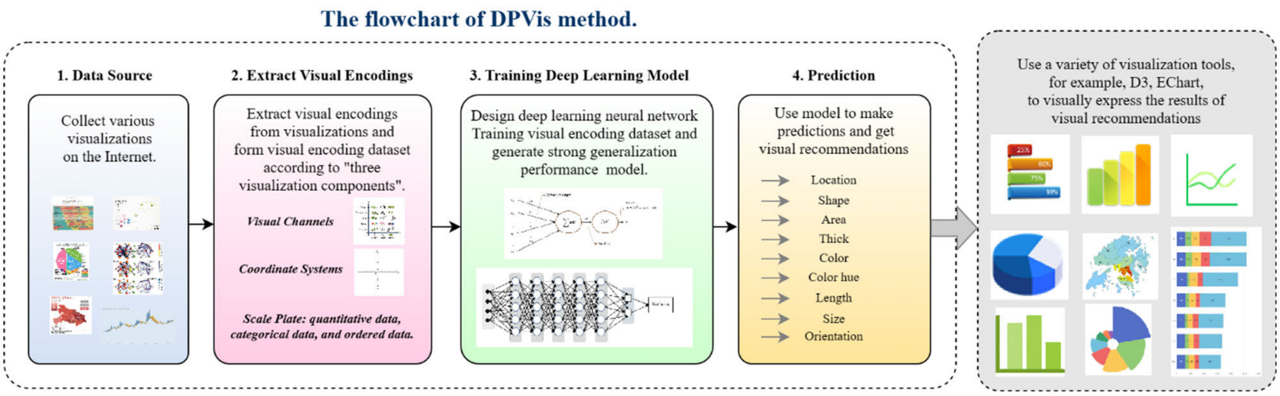


FIGURE 1. Owchart of DPVis method.

mapping, that is, the mapping relationship between data and visualization results, which can encourage readers to obtain information from graphics quickly. Therefore, we can regard data visualization as a combination of graphic elements bearing visual encodings.

Visual channels and visual markers make up the two components of visual encodings orthogonal, as shown in Fig. 2. Orthogonality means that in this case, every visual channel can be used to map to any marker, from which visual channels for visual encoding can subsequently be constructed. Markers are usually geometric elements used to illustrate how data are categorized according to their nature, such as points, lines, faces, and bodies. As shown on the X-axis in Fig. 2. As indicated on the Y-axis in Fig. 2, visual channels are used to quantitatively characterize the presentation state of markers in graphics and manage their visual features, which primarily comprise location, size, shape, texture, direction, color and so on [19]. Currently, the visual channels have now been expanded to include length, area, volume, hue, saturation, transparency, brightness, blurring, focusing, etc.

Visualization is the process of converting many types of data into understandable, simple-to-understand, and easy-to-remember visual channels using the principles of visual encoding. Therefore, finding the appropriate visual encodings is equivalent to finding the appropriate visual channels for data output since the appropriate visual channels are used for the visual encodings.

b: COORDINATE SYSTEMS

The data should be arranged and the graphic location or coordinate system must be specified when visually encoding information. There are many different types of coordinate systems, but three are frequently used in visual analysis: the polar coordinate system, the geographical coordinate system, and the Cartesian coordinate system.

Cartesian coordinate system: The X and Y-axes intersect vertically, which is very common, as shown in Fig. 3(a).

Polar coordinate system: A coordinate system based on the radius and angle, which can be understood as a coordinate system formed by rotating the X-axis by 360°, as shown in Fig. 3(b). A polar coordinate system is employed in visualization graphics such as pie charts and radar charts.

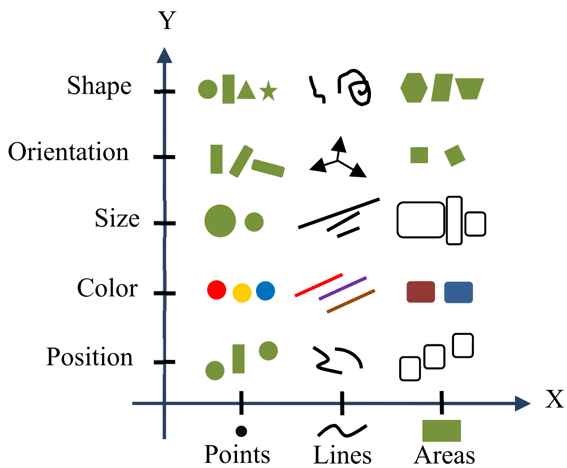


FIGURE 2. Visual encodings [19].

Geographic coordinate system: The geographic coordinate system is used to identify geographic location information using longitude and latitude. In visualization, maps are widely used to represent geographic data.

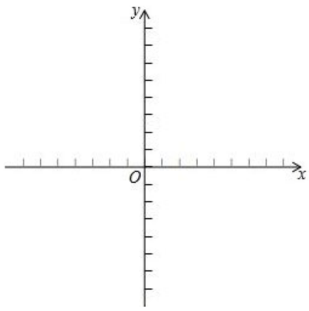
c: SCALE PLATE

The coordinate system indicates how many dimensions can be visualized, whereas the scale plate indicates the data types in each dimension, which are usually divided into three fundamental data categories: quantitative data, categorical data, and ordered data, as follows:

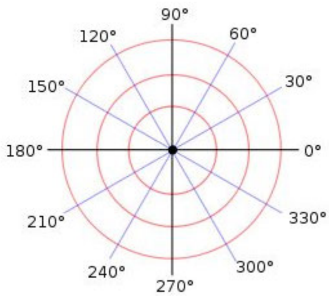
Quantitative data, such as percentages, is used to describe the value of the data.

Categorical data is used to describe how qualities are used to classify data.

Ordered data is used to describe how the data changes over time.



(a) Cartesian coordinate system



(b) Polar coordinate system

FIGURE 3. Coordinate system.

TABLE 1. Common visual channels and fundamental data priorities (from high to low).

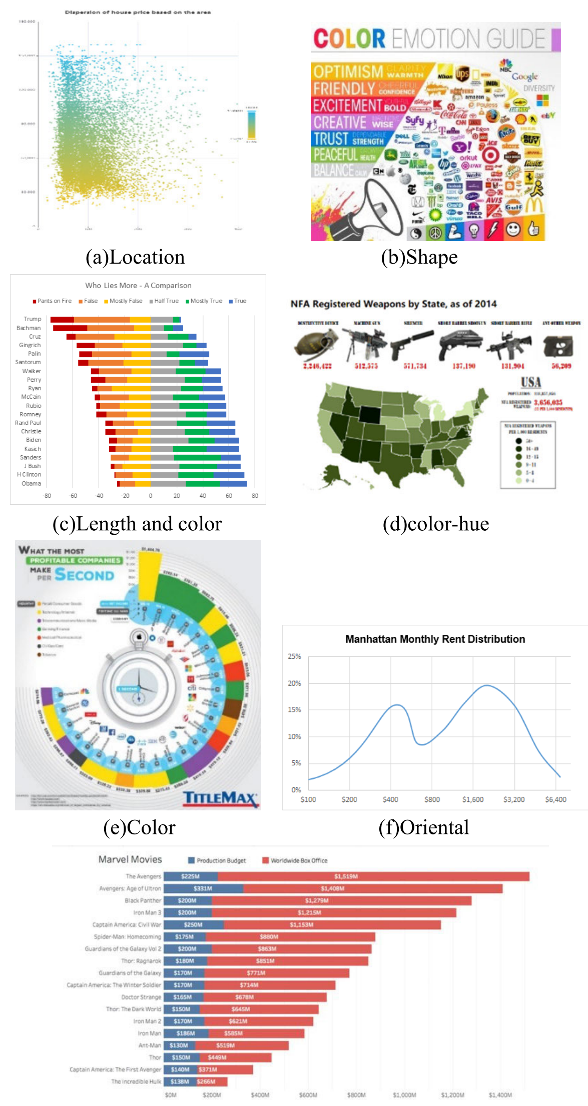
Categorical	Ordered	Quantitative
Location	Location	Location
Size	Area	Size
Shape	Color and color hue	Area
	Shape	Color and color hue
		Orientation

2) VISUAL ENCODING PRINCIPLE

Users may accurately decode the information contained in visual encodings when many visual channels are combined logically [41]. However, assuming that the data have n dimensions and there are m visual channels to choose from in accordance with the data properties, there are $(n+1)^m$ encoding schemes, and it is highly challenging to choose the optimal one among them.

Table 1 summarizes the priorities of the visual channels utilized for the visual encoding of various data types. Fig. 4(b) illustrates how categorical and ordered data may be encoded using shape, whereas Fig. 4(c) and (g) illustrate how quantitative data can be encoded using length. As illustrated in Fig. 4(a), (d), and(e), the location, color, and color hue may encode any type of data.

Visual encoding requires the use of the fewest number of visual channels. The visual system becomes muddled and overlapping if there are too many visual channels. Information density, visual prominence, and expressiveness should also be taken into account. Consequently, visual encoding must adhere to the design guidelines listed in Table 2.



(a)Location (b)Shape (c)Length and color (d)color-hue (e)Color (f)Oriental (g)Length and color

FIGURE 4. Various visual encodings.

3) VISUAL ENCODING DATASET

We analyzed the collected graphics, extracted three essentials of visualization from them, and annotated them to create a visual encoding dataset.

First, the coordinate system was extracted. In most cases, we created visualization using Cartesian coordinates system, which is appropriate for the majority of data types. Most of the graphics collected for our study were designed in Cartesian coordinates. We can also display data along the Z-axis, but Table 2 demonstrates that the effect is not favorable according to the empirical principle of visual encoding.

Second, we extract the scale plate of the graphics, that is, the type of encoded data. We divided the data into four categories—spatial data, temporal data, spatial-temporal data, and general data—covering almost all of the common types of data to enhance the ability of dataset to be explained.

Third, we extracted of the visual channels encoded in the graphics as the data features of this record. Then, as the values

TABLE 2. Empirical principles of visual encoding.

Error	Correction	Experience Summary
Confusing pie chart segmentation	Place the largest part in a clockwise position at 12 o'clock.	A pie chart is best divided into no more than five pieces. The largest part is at 12 o'clock.
Use inconsistent lines in a line chart	Use solid lines.	Dotted lines are easy to distract. Try not to use them.
Data classification and sorting confusion	Sort in alphabetical order.	Sort the data categories in alphabetical order, size order, or value.
Obscure data	Add encoding methods such as transparency and trend curve.	Make the chart as easy as possible to understand the data.
Too many different colors used	Use Hue.	Too much color will add unbearable weight to the data. It is better to use no more than five colors. If there are more than five colors, the same color system or analog color should be used.
3D effect	Use 2D.	In visual encoding design, the most important principle is; the simpler is the better.

of the X and Y-axis dimension attributes in this record, the data types encoded by the X and Y-axes in the coordinate system are extracted.

Finally, the visual channels used in the graphics were extracted as the supervision label of this piece of data, thus forming a supervised visual encoding dataset. It is also evident that a scale plate, a coordinate system, and visual channels are used in the traditional method of visual encoding.

There are 207 records in the visual encoding dataset to be trained. Each item of dataset consists of four attributes: data source, X-axis encoded data type, Y-axis encoded data type, and data type. The label of this item is the visual channel. The visual channels, which are frequently utilized in visualization graphics, were employed in this study. The nine categories into which we split the labels were location, size, length, shape, thickness, area, color, color hue, and orientation.

D. METHODS

This section explains the deep-learning-based visual encoding model training procedure. Quick and precise automatic visual encoding includes data pretreatment, model construction, model training and testing. These are steps 3 and 4 of the workflows in Fig. 1.

The main principle of deep learning model training is as follows. First, an initial value can be initialized randomly or be set according to previous experience, and the anticipated outcome can then be determined. The difference between the actual value in the training set marked beforehand and the predicted outcome can then be compared. Then, the model can be adjusted to approach the correct direction based on the basis of repeated, until the predicted result is almost the same as the real value.

The following stages may be used to summarize the deep learning model training process:

1. Establish a deep learning model with a variety of learnable weight parameters;
2. Define that loss function;
3. Calculate the loss value, which is the distinction between the output value and the target value;
4. Backpropagation gradient into the parameters of the deep learning model;
5. Adjust the weight value of the network model in line with the update rules;
6. Iterative calculations are used to reduce the loss on the input.

1) FEATURE PROCESSING

The visual encoding dataset we constructed cannot be directly applied to train the model. Therefore, preprocessing is necessary. First, we numerically processed all fields in the visual encoding dataset, changing the original type values into numeric values, and putting the dataset into a deep learning model-friendly structure. The data were divided into 70/20/10 training/validation/testing sets to ensure that there was no overlap between the three sets. Using 5-fold cross-validation, the model was trained and tested five times. So, the study results that were provided were an average of the results from the five testing sets. The model parameters were trained and adjusted using the training set. The correctness of the model is checked, and its hyperparameters are modified using the verification set. The capacity of the model for generalization was confirmed using a testing set. The training and verification sets should be kept apart so that the model's performance on the verification set can reflect its generalizability.

2) MODEL PREPARATION

In this section, we designed the deep learning model's layer count, dimension matching between layers, and selection of proper activation function, loss function, and associated parameters.

First, we defined the hidden layer size and the depth of the neural network. Our model was a fully connected feed-forward neural network with three hidden layers. Each layer was composed of neurons with ReLU activation functions, as shown in (1), and the model was implemented by using PyTorch [42].

$$f(x) = \max(0,x) \tag{1}$$

The Softmax function must process the output layer results before the final model results can be achieved, as illustrated in (2). The multiclassification output sequences may be converted into relative probabilities using the Softmax function, which maps the output neurons to real values between 0 and 1 and guarantees that the normalized guaranteed sum is 1.

$$S_i = \frac{e^{V_i - D}}{\sum_i^C e^{V_i - D}} \tag{2}$$

where V_i is the outcome of the classifier's previous outcome unit and $D = \max(V_i)$. The total number of categories is C, where i represents the category index. S_i represents the proportion of the current element index to the total number of elements.

The cross-entropy loss function is widely used in classification scenarios since data training is a classification problem. The cross-entropy loss function is presented in (3). Cross-entropy is a measure of how far apart two probability distributions are from one another. The cross entropy decreases with increasing proximity.

$$Loss = -\sum_i t_i ln y_i \tag{3}$$

where t_i represents the output value and y_i represents the value processed by the Softmax function. Gradient dissipation in classification issues can be minimized by using cross-entropy as the loss function.

Finally, the loss function was iterated and reverse updated using the gradient descent optimizer, as shown in (4).

$$x \leftarrow x - \eta f'(x) \tag{4}$$

where η is the learning rate, which is a manually adjustable hyperparameter that can significantly impact the algorithm's precision and effectiveness. The learning rate was initialized at 1×10^{-1},

3) MODEL TRAINING

The training set served as the input. The input layer of the neural network receives the input for each record, and through the process of forward propagation, calculates the output value of each neuron in the output layer.

The error of the output layer was calculated before using the backpropagation algorithm to obtain the error of each neuron in each layer.

The connection weight ω and threshold θ of each neuron can be obtained by the error and then multiplied by the negative learning rate $(-\eta)$ to obtain $\Delta\omega$ and $\Delta\theta$, and the ω and θ of each neuron are updated to $\omega+\Delta\omega$ and $\theta+\Delta\theta$.

It is necessary to repeat this entire training process hundreds, perhaps even tens of thousands of times. We can create a neural network model with a pretty low loss after training.

During the neural network learning process, the "connection weight" between neurons and the threshold value of each functional neuron are both altered in line with the training data. In other words, a neural network "learns" based on the connection weight and threshold parameters [43].

Cross-validation is another method used in deep learning to train a model. To validate the state and convergence of the model, a validation set was set aside throughout the model training phase. Discover which group of hyperparameters performs best based on the performance of several groups of model verification sets to determine the appropriate number of hidden layers and alter the hyperparameters. Overfitting of the model during training may also be monitored simultaneously using the verification set at the same time. In general, overfitting occurs because if training is continued after the performance of the verification set has stabilized, the performance of the training set will continue to climb, but the performance of the verification set will decline instead of increase. As a result, the stopping point of the backpropagation algorithm is determined using the verification set.

The trained model was then tested on the testing set, and its performance was assessed in accordance with the error, allowing the accuracy of the model to be determined [44].

The deep neural network iterated on our training set and its loss function, as shown in Fig. 5, steadily decreased, and tended to be stable, realizing the goal of accurately identifying the visual encoding dataset.

The prediction was produced on the testing set once the model had been trained. The accuracy rate of the testing set, which was used to assess the generalizability of the final model, reached 92.01%. The results demonstrate the effectiveness of the model and its potential generalizability.

4) PRESERVATION AND APPLICATION OF THE MODEL

The parameters that the model learned were then kept for later use in practice when it developed a trained model with good generalization performance. The purpose of the visual encoding model we developed was to forecast the visual channels, enabling users to realize automatic visual channel suggestions in the visual tools and finish the visual design quickly and automatically [45].

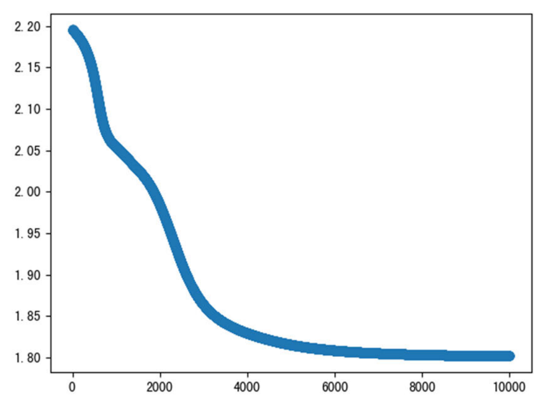


FIGURE 5. Training loss function of the visual encoding dataset.

IV. EVALUATING PERFORMANCE

To illustrate the potential value of DPVis, we compared DPVis with two other machine learning approaches: K-Nearest Neighbor (KNN) and Naive Bayes (NB). To show that our method is more efficient, convenient and practical. All models were implemented with scikit-learn [46] and default parameters, and the visual encoding dataset was

IEEE Access

trained with the same proportion of training and testing sets and its accuracy was obtained, as shown in Fig. 6. The random parameter search of NB and KNN models did not lead to significant performance improvement. The results show that DPVis method has higher accuracy than other machine learning methods. It has good applicability to the visual encoding dataset constructed by us.

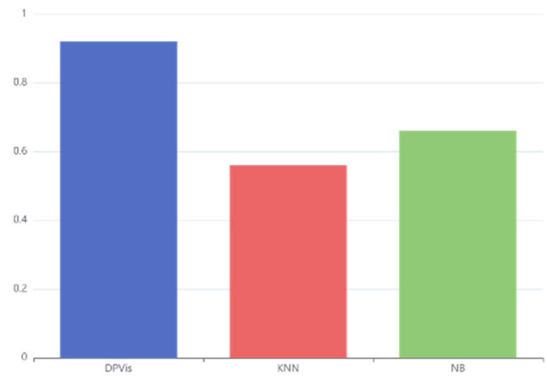


FIGURE 6. Accuracy comparison.

V. DISCUSSIONS

Tens of thousands of more iterative calculations on the input dataset are required to train the model. The training procedure is essentially a self-learning process that involves continual iterative updating of the weight parameter until the average loss error reaches the lowest point and improves performance.

Furthermore, the performance and speed of convergence of the model were significantly influenced by the initialization of the weight parameters. Gradient disappearance or gradient explosion in the process of gradient descent is common as the number of neural-network layers increases. These two issues may be solved with a good initialization of the weight parameter, which is also beneficial for the performance and speed of convergence of the model.

The data sample used for model fitting was known as the training set. The weight parameters were trained throughout the training procedure using gradient descent of the training error. A validation set is a distinct set of datasets aside during model training that can be used to modify the model's hyperparameters and provide a preliminary assessment of the model's performance. The training procedure used verification set. After several iterations of training, a verification set is often used to evaluate the training effect, which has the following benefits: the model may diverge from the verification set or provide endless results. For example, flaws in the model's parameters might be discovered over time. In this case, the training may be stopped early so that the parameters or model can be changed without having to wait until the training is complete. Second, it is possible to confirm the model's capacity for generalization. The overfitting of the model should be considered if the effect on the verification set differs significantly from that on the training set. Third, verification sets can be used to evaluate various models. For example, an ideal network depth, the stopping point of the backpropagation algorithm, or the number of hidden layer neurons in a neural network can all be determined using a verification set.

The depth and breadth of deep-learning neural networks may be adjusted at will. Theoretically, it is possible to map any function. Deep learning is hence capable of resolving complicated issues. Many systems are compatible with frameworks such as PyTorch and TensorFlow.

Our approach can be used as a workflow extension for current automatic visualization techniques in particular visualization applications. Designers can remark on their data features, X-axis encoded data features, Y-axis encoded data features, and this data label when designing visualization graphics. Immediately following creation, a supervised visual encoding record was saved. To expand their learning coverage and improve performance, visualization tools may use these datasets, as more users produce graphics and learn from their visual encodings and rules.

VI. CONCLUSION

In this paper, we describe DPVis, a deep learning approach for automatic visualization. The visualization process was turned into a deep learning modeling process in this study, and the visual encoding dataset was created by extracting the three components of visualization from the existing visualization graphics. Automatic visualization is then achieved using the visual encoding model that is learned from this dataset, which is expected to improve the intelligence of visual analysis [47].

The accuracy of our method surpasses that of others, and it is capable of quickly and intelligently realizing visual encoding without the need for specialized experience or knowledge. This is a practical technique that integrates deep learning and artificial intelligence into the visual analysis process.

We are aware of the limitations of the relatively few and narrowly focused visualization graphics that we collected. Although we collected the most prevalent graphics from different types of data and performed a preliminary analysis, we also removed duplicate graphics, which might have resulted in a certain proportional imbalance. For training using other visualization tools, there is a requirement for a broader diversity of graphics on the data side.

Our dataset is currently downloaded from the Internet and annotated manually, which is laborious, time consuming, and costly. Consequently, the next research focus of this study will be on efficiently, automatically, and correctly producing a visual encoding dataset. There are also automatic extraction approaches [48], [49], [50] and applications [51], [52] to extract visual encodings from graphics, although visual encodings can be inaccurate and do not conform to the practice of extracting visual encodings according to the ''three essentials of visualization'' in this study, particularly the number of professional visual channels related to various fields continues to increase. Future considerations might also include the extraction of polar and geographic coordinates, improvement of the scale plate, and the extraction and annotation of interactive analysis.

REFERENCES

[1] R. V. Guimarães, "Visual encoding quality and scalability in information visualization," Univ. Ontario Inst. Technol., Oshawa, ON, Canada, Tech. Rep., 2019.
[2] G. Federico, F. Osiurak, M. A. Brandimonte, M. Salvatore, and C. Cavaliere, "The visual encoding of graspable unfamiliar objects," *Psychol. Res.*, vol. 87, pp. 452–461, Mar. 2022.
[3] M. A. Kuhail, S. Farooq, R. Hammad, and M. Bahja, "Characterizing visual programming approaches for end-user developers: A systematic review," *IEEE Access*, vol. 9, pp. 14181–14202, 2021.
[4] S. Zhu, G. Sun, Q. Jiang, M. Zha, and R. Liang, "A survey on automatic infographics and visualization recommendations," *Vis. Informat.*, vol. 4, no. 3, pp. 24–40, Sep. 2020.
[5] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
[6] 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," *Vis. Informat.*, vol. 2, no. 2, pp. 136–146, 2018.
[7] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 341–350, Jan. 2017.
[8] H. Mei, W. Chen, Y. Ma, H. Guan, and W. Hu, "VisComposer: A visual programmable composition environment for information visualization," *Vis. Informat.*, vol. 2, no. 1, pp. 71–81, 2018.
[9] R. López-Cortijo, J. G. Guzmán, and A. A. Seco, "iCharts: Charts for software process improvement value management," in *Proc. Eur. Conf. Softw. Process Improvement*. Springer, 2007, pp. 124–135.
[10] M. Mauri, T. Elli, G. Caviglia, G. Uboldi, and M. Azzi, "RAWGraphs: A visualisation platform to create open outputs," in *Proc. 12th Biannual Conf. Italian SIGCHI*, Sep. 2017, pp. 1–5.
[11] N. J. Pioch and J. O. Everett, "POLESTAR: Collaborative knowledge management and sensemaking tools for intelligence analysts," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage.*, 2006, pp. 513–521.
[12] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager: Exploratory analysis via faceted browsing of visualization recommendations," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 649–658, Jan. 2016.
[13] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager 2: Augmenting visual analysis with partial view specifications," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2017, pp. 2648–2659.
[14] D. Ren, T. Höllerer, and X. Yuan, "iVisDesigner: Expressive interactive design of information visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2092–2101, Dec. 2014.
[15] S. F. Roth, J. Kolojejchick, J. Mattis, and M. C. Chuah, "SageTools: An intelligent environment for sketching, browsing, and customizing data-graphics," in *Proc. Conf. Companion Hum. Factors Comput. Syst. (CHI)*, 1995, pp. 409–410.
[16] D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in *Proc. 14th Int. Conf. Intell. User Interfaces*, Feb. 2009, pp. 315–324.
[17] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang, "Text-to-viz: Automatic generation of infographics from proportion-related natural language statements," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 906–916, Jan. 2020.
[18] Y. Chen, S. Barlowe, and J. Yang, "Click2Annotate: Automated insight externalization with rich semantics," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, Oct. 2010, pp. 155–162.
[19] B. Jacques, *Semiology of Graphics: Diagrams, Networks, Maps*. Madison, WI, USA: Univ. of Wisconsin Press, 1983.
[20] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, "Foresight: Recommending visual insights," 2017, *arXiv:1707.03877*.
[21] H. Ehsan, M. A. Sharaf, and P. K. Chrysanthis, "MuVE: Efficient multi-objective view recommendation for visual data exploration," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 731–742.
[22] A. Key, B. Howe, D. Perry, and C. Aragon, "VizDeck: Self-organizing dashboards for visual analytics," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2012, pp. 681–684.
[23] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov./Dec. 2007.
[24] E. C. Alexander, C.-C. Chang, M. Shimabukuro, S. Franconeri, C. Collins, and M. Gleicher, "Perceptual biases in font size as a data encoding," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2397–2410, Aug. 2018.
[25] D. A. Szafir, "Modeling color difference for visualization design," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 392–401, Jan. 2018.
[26] Y. Kim and J. Heer, "Assessing effects of task and data distribution on the effectiveness of visual encodings," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 157–167, Jun. 2018.
[27] D. B. Perry, B. Howe, and C. Aragon, "VizDeck: Streamlining exploratory visual analytics of scientific data," Tech. Rep., 2013.
[28] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis, "SEEDB: Automatically generating query visualizations," Tech. Rep., 2014.
[29] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Towards a general-purpose query language for visualization recommendation," in *Proc. Workshop Hum.-In-the-Loop Data Anal.*, Jun. 2016, pp. 1–6.
[30] Y.-R. Cao, J.-Y. Pan, and W.-C. Lin, "User-oriented generation of contextual visualization sequences," in *Proc. Extended Abstr. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–8.
[31] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2019, pp. 1–12.
[32] J. McAuley, R. Goel, and T. Matthews, "ExploroBOT: Rapid exploration with chart automation," in *Proc. VISIGRAPP* , 2019, pp. 225–232.
[33] V. Dibia and C. Demiralp, "Data2Vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE Comput. Graph. Appl.*, vol. 39, no. 5, pp. 33–46, Sep. 2019.
[34] H. Lin, D. Moritz, and J. Heer, "Dziban: Balancing agency & automation in visualization design via anchored recommendations," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–12.
[35] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang, "DataShot: Automatic generation of fact sheets from tabular data," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 895–905, Jan. 2020.
[36] T. Alhersh, H. Stuckenschmidt, A. U. Rehman, and S. B. Belhaouari, "Learning human activity from visual data using deep learning," *IEEE Access*, vol. 9, pp. 106245–106253, 2021.
[37] Y. Luo, X. Qin, N. Tang, and G. Li, "DeepEye: Towards automatic data visualization," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Apr. 2018, pp. 101–112.
[38] C. Qian, S. Sun, W. Cui, J.-G. Lou, H. Zhang, and D. Zhang, "Retrieve-then-adapt: Example-based automatic generation for proportion-related infographics," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 443–452, Feb. 2021.
[39] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, "Calliope: Automatic visual data story generation from a spreadsheet," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 453–463, Feb. 2021.
[40] Y. Kim and J. Heer, "Gemini: A grammar and recommender system for animated transitions in statistical graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 485–494, Feb. 2021.
[41] M. Chen, S. Walton, K. Berger, J. Thiyagalingam, B. Duffy, H. Fang, C. Holloway, and A. E. Trefethen, "Visual multiplexing," *Comput. Graph. Forum*, vol. 33, no. 3, pp. 241–250, Jun. 2014.
[42] A. Paszke et al., "Automatic differentiation in Pytorch," Tech. Rep., 2017.
[43] Z.-H. Zhou, *Machine Learning*. Springer, 2021.
[44] Y.-C. Chang, C.-H. Ku, and D.-D.-L. Nguyen, "Predicting aspect-based sentiment using deep learning and information visualization: The impact of COVID-19 on the airline industry," *Inf. Manage.*, vol. 59, no. 2, Mar. 2022, Art. no. 103587.
[45] J. Wang, Z. Mo, H. Zhang, and Q. Miao, "A deep learning method for bearing fault diagnosis based on time-frequency image," *IEEE Access*, vol. 7, pp. 42373–42383, 2019.
[46] F. Pedregosa, S. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Dec. 2011.
[47] V. Capocasale and G. Perboli, "Standardizing smart contracts," *IEEE Access*, vol. 10, pp. 91203–91212, 2022.
[48] B. Tang, S. Han, M. L. Yiu, R. Ding, and D. Zhang, "Extracting top-K insights from multi-dimensional data," in *Proc. ACM Int. Conf. Manage. Data*, May 2017, pp. 1509–1524.
[49] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko, "Augmenting visualizations with interactive data facts to facilitate interpretation and communication," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 672–681, Jan. 2019.

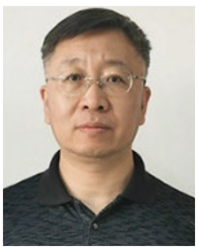
[50] J. Heer, T. Ropinski, and J. van Wijk, ''Reverse-engineering visualizations: Recovering visual encodings from chart images,'' *Population*, vol. 3, no. 1, p. 570, 2017.

[51] J. Poco, A. Mayhua, and J. Heer, ''Extracting and retargeting color mappings from bitmap images of visualizations,'' *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 637–646, Jan. 2018.

[52] J. E. Zhang, N. Sultanum, A. Bezerianos, and F. Chevalier, ''DataQuilt: Extracting visual elements from images to craft pictorial visualizations,'' in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–13.

JIA LIU was born in Tianjin, China, in 1984. She received the B.S. degree in communication engineering from the Tianjin University of Commerce, in 2007, and the M.S. degree in software engineering from the Beijing University of Technology, in 2010. She is currently pursuing the Ph.D. degree in information and communication engineering with Space Engineering University.

From 2010 to 2013, she was a Lecturer with the Information University of China. Since 2014, she has been a Software Engineer and has developed several software applications. Her current research interests include high-dimensional spatiotemporal data visualization and visual analysis, digital earth, and big data analysis.



GANG WAN was born in Hebei, China, in 1972. He received the Ph.D. degree in cartography and geographic information engineering, in 2006.

He is currently a Professor with the School of Space Information, Space Engineering University, the Director of the Department of Surveying and Mapping and Space Environment, and a Ph.D. Advisor. He is the author of more than ten books, more than 100 articles, and more than 20 inventions. His current research interests include virtual geographic environment, spatiotemporal big data, and spatial environment exploration.

Dr. Wan was a member of the System Simulation Society's Simulator Professional Committee and the Society of Surveying and Mapping's Professional Committee for Cartography and Geographic Information Engineering, in 2011. He received two second-place awards: the National Teaching Achievements Award and the National Science and Technology Progress Award. He was elected for the Ministry of Education's New Century Outstanding Talents Support Program, in 2008. He received the Henan Province Youth Science and Technology Award, in 2010, the 2013 Xia Jianbai Surveying and Mapping Entrepreneurship and Technological Innovation Award, and the National ''University GIS Innovation Talent Award,'' in 2014. He was selected as the Special Expert of the Ministry of Civil Affairs' Key Laboratory for Disaster Assessment and Risk Prevention, in 2012.

PANKE DENG received the M.S. degree in mechanical engineering from Beijing Jiaotong University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree with Space Engineering University. His main research interests include information communication, remote sensing, deep learning, and visual analysis.

CHU LI received the B.S. degree in navigation engineering from Space Engineering University, Beijing, China, in 2022, where he is currently pursuing the master's degree in information and communication engineering. His main research interests include visualization, mars remote sensing, and deep learning.

SHUAI WANG received the Ph.D. degree in communication and information systems from Space Engineering University, in 2018. He is currently a Lecturer with Space Engineering University. His research interest includes three-dimensional imaging.

YUNXIA YIN was born in Shaanxi, China, in 1974. She received the M.S. degree in software and network engineering from the Beijing University of Technology, in 2012. She is currently an Associate Professor with Space Engineering University. Her research interests include information visualization, big data visual analysis, artificial intelligence, and computer vision.

LEI LIU was born in Hebei, China, in 1984. He received the M.S. and Ph.D. degrees in remote sensing from Peking University, China, in 2011 and 2018, respectively. He is currently an Associate Professor with Space Engineering University. His research interests include geographic information systems, remote sensing, and computer vision.

YAO MU received the Ph.D. degree in environmental science and engineering from the China University of Geosciences.

Since 2019, she has been a Lecturer with Space Engineering University. She is the author of more than ten articles. Her current research interests include space weather forecast and spacecraft microbiome.

Dr. Mu received the Outstanding Graduate Student Award, in 2014 and 2015, the Top Academic Scholarship for five years running, and the National Scholarship for graduate student, in 2017.