

# EL-VIT: Probing Vision Transformer with Interactive Visualization

Hong Zhou, Rui Zhang, Peifeng Lai, Chaoran Guo, Yong Wang\*, Zhida Sun and Junjie Li

**Abstract**—Nowadays, Vision Transformer (ViT) is widely utilized in various computer vision tasks, owing to its unique self-attention mechanism. However, the model architecture of ViT is complex and often challenging to comprehend, leading to a steep learning curve. ViT developers and users frequently encounter difficulties in interpreting its inner workings. Therefore, a visualization system is needed to assist ViT users in understanding its functionality. This paper introduces EL-VIT, an interactive visual analytics system designed to probe the Vision Transformer and facilitate a better understanding of its operations. The system consists of four layers of visualization views. The first three layers include model overview, knowledge background graph, and model detail view. These three layers elucidate the operation process of ViT from three perspectives: the overall model architecture, detailed explanation, and mathematical operations, enabling users to understand the underlying principles and the transition process between layers. The fourth interpretation view helps ViT users and experts gain a deeper understanding by calculating the cosine similarity between patches. Our two usage scenarios demonstrate the effectiveness and usability of EL-VIT in helping ViT users understand the working mechanism of ViT.

**Index Terms**—Vision Transformer, Education Tool, Explainable AI, Visual Analysis

## I. INTRODUCTION

Deep learning has become an integral part of our daily lives, with its application spanning across various industries. While deep learning models have continuously improved in terms of performance, they have also grown in complexity, which raises the learning curve. This paper focuses on the Vision Transformer (ViT) model, a deep learning architecture that incorporates principles from natural language processing into computer vision. ViT [1] has demonstrated its superiority over Convolutional Neural Networks (CNN) in various computer vision downstream tasks, including image classification, object detection, semantic segmentation, and image captioning. Consequently, the ViT model has generated significant interest among practitioners, students, and experts, prompting many individuals to explore this technology. However, for novices, understanding the inner workings of the ViT model can be challenging, given its intricate layer structures, inter-layer data transmission methods, convolution, and multi-head attention mechanisms. Taking image classification as an example, beginners may struggle to comprehend the journey from initial

image data to the ultimate classification prediction. Similarly, computer vision students may feel difficult to understand concepts such as Query, Key, and Value, and experts may also be perplexed by the inner workings of ViT and the principles underlying its classification. Therefore, the development of a visualization system to facilitate the understanding and application of this technology for both beginners and experts is important.

To ensure a positive user experience with the visualization system, we encountered several challenges during the design of the visualization tool. Firstly, the layer structure of ViT is highly intricate, accompanied by a vast number of parameters. It not only possesses a deep layer structure (e.g., the simplest ViT model, ViT-B/16, comprises 12 Transformer blocks), but also incorporates various types of layers (such as Query, Key, Value, Glue, etc.), each involving distinct mathematical operations. Comprehensive guidance is necessary to assist users in gaining a thorough understanding of the ViT model. Secondly, existing visualization tools lack a unified overview of educational aids and interpretability explanations. Nowadays, numerous visualization systems employ interactive visualizations to assist users in exploring and learning deep learning models, such as CNN Explainer [2] and DNN Genealogy [3], which respectively aid beginners in understanding convolutional neural networks and deep neural networks. Additionally, some visualization systems are dedicated to providing interpretability explanations for deep learning models, aiding experts in understanding and diagnosing these models [4], [5]. Both aspects are crucial, and EL-VIT is proposed to address these challenges with the aim of facilitating the understanding and learning of the ViT model for both experts and non-experts.

The major contributions of this paper are as follows:

- **We introduce an interactive visualization tool, EL-VIT, to help ViT learners and users to explore ViT models.** Currently, many visualization tools cater to experts, with limited availability of educational-focused visualization tools. The proposed visualization platform in this paper aims to fill this research gap and serves educational purposes.
- **We utilize a multi-view visualization system design, each view providing unique insights into the model.** This approach fosters a holistic comprehension of the ViT model and promotes the acquisition of model-related knowledge. EL-VIT is web-based, requiring no backend programming or the need for users to install specific software. Users can explore the operations of ViT simply

\*Corresponding author.

• Hong Zhou, Zhida Sun and Junjie Li are with Shenzhen University, E-mail: {hzhou|zhida.sun|jj.li}@szu.edu.cn

• Rui Zhang, Peifeng Lai and Chaoran Guo are with Shenzhen University, E-mail: {zhangrui2021|laipeifeng2022|guochaoran2021}@email.szu.edu.cn

• Yong Wang is with Singapore Management University, E-mail: yong-wang@smu.edu.sg

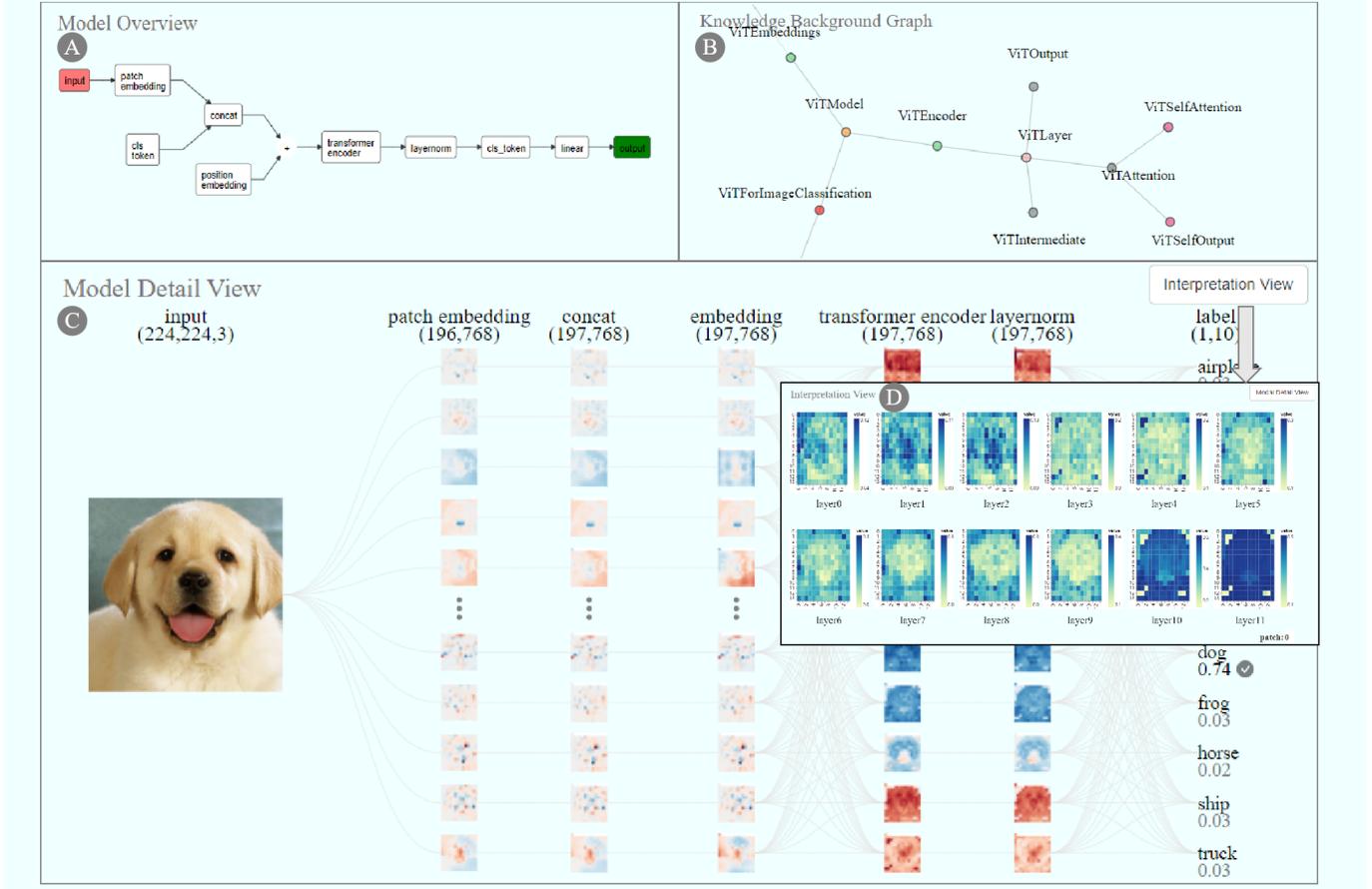


Fig. 1. The overview diagram of the visualization system.

by accessing the web page. EL-ViT also incorporates various interactive visualization methods, providing users with a seamless experience and aiding in their better comprehension of ViT.

- **We offer an innovative approach to enhancing the interpretability of the ViT model.** Rather than concentrating on visualizing attention weights, EL-ViT calculates cosine similarity for the outputs of each Transformer block. It reveals that patches corresponding to the same object tend to exhibit higher similarity, and the token used for classification, CLS token, often demonstrates greater similarity to the patches associated with the classified objects. By providing interpretation view, EL-ViT facilitates convenient model exploration for users.

## II. BACKGROUND

Within this section, we present a concise retrospective analysis of the Vision Transformer model, delivering a succinct exposition of the foundational principles and technical specifics of the ViT model. This segment lays the groundwork for the subsequent visualization chapter of this paper.

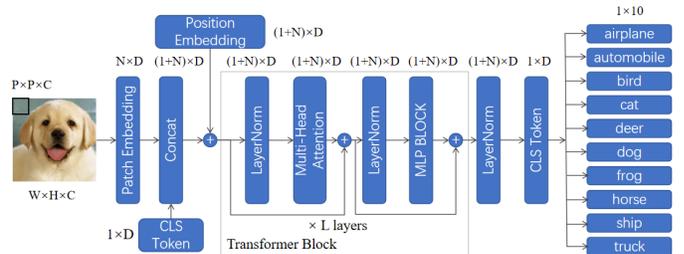


Fig. 2. The architectural diagram of the ViT-B/16 model.

Transformer [6] has achieved significant success in the field of natural language processing, leading to its application in computer vision. On large-scale datasets, ViT has outperformed CNN and demonstrated favorable transferability in downstream tasks. Specifically, as illustrated in Fig. 2, within the ViT model, when provided with an input image  $X \in R^{H \times W \times C}$ , where  $H$ ,  $W$ , and  $C$  represent the image's height, width, and number of channels, respectively, the image is divided into  $N$  patches. Any patch's shape is  $P \times P \times C$  with  $N = HW/P^2$ . The number of convolutional kernels

is denoted as  $D$ . Subsequently, a linear transformation is performed using  $D$  convolutional kernels, each of shape is  $P \times P \times C$ , applied to  $N$  patches, resulting in a two-dimensional matrix. This matrix is concatenated with a CLS token, and position encoding is added. This process yields the embedding layer, as illustrated in Equation 1.  $z_0$  precisely corresponds to the input of the Transformer encoder. The Transformer encoder consists of  $L$  layers of Transformer blocks, each of which comprises various distinct layers. These layers encompass LayerNorm, Multi-Head Self-Attention (MSA), and Multi-Layer Perceptron (MLP) blocks. Prior to each block, LayerNorm (LN) is applied, and after each block, residual connections are introduced, as explicitly defined in Equations 2 and 3. Following these steps, the results are extracted with 0 dimension, and the CLS token is obtained. Subsequently, layer normalization, as demonstrated in Equation 4, yields the output  $y$ , where  $z_L^0$  represents the CLS token. The dimensionality of  $y$  is  $1 \times D$ . Finally, classification predictions are obtained through a linear transformation.

$$z_0 = [x_{cls}; x_p^1 E; x_p^2 E; \dots; x_p^N E] + E_{pos}, E_{pos} \in R^{(N+1) \times D} \quad (1)$$

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1}, l = 1 \dots L \quad (2)$$

$$z_l = MLP(LN(z'_l)) + z'_l, l = 1 \dots L \quad (3)$$

$$y = LN(z_L^0) \quad (4)$$

Within the ViT forward process, the attention mechanism assumes a crucial role. As outlined in Fig. 2, it resides within the multi-head attention layer of the Transformer block. Following normalization in the preceding layer, the resultant matrices are subject to separate multiplications with  $W^Q$ ,  $W^K$ , and  $W^V$ , yielding Q, K, and V. Consequently, self-attention can be computed:

$$Attention(Q, K, V) = A \cdot V = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (5)$$

Within this paper,  $A \in R^{N \times N}$  denotes the attention weights, which determine the significance of relationships between different positions in the input sequence. Given that this paper employs the ViT-B/16 model, it follows that  $H = W = 224$ ,  $C = 3$ ,  $P = 16$ ,  $D = 768$ , and  $N = 196$ .

### III. RELATED WORK

#### A. Visualization for Deep Learning Education

As researchers continually develop visualization systems, the number of visualization systems designed for educational purposes has gradually increased. These systems aim to assist beginners in better understanding concepts related to deep learning models. For example, the web-based MNIST demo on ConvNetJS [7] dynamically illustrates the training process and intermediate results of digit recognition. Teachable

Machine [8] is a web-based GUI tool that enables users to learn about machine learning classification by creating and using their own classification models. DNN Genealogy [3] serves as an educational tool that provides visual analytical guidance for understanding and applying deep neural networks through interviews, surveys, and summarization and analysis of 66 representative DNNs. TensorFlow Playground [9] allows users to experiment with neural networks through direct manipulation, making it accessible to users unfamiliar with deep learning. GAN Lab [10] enables interactive training of generative models for learning and experimenting with adversarial neural networks. CNN Explainer [2] uses real image data to explain the model structure and mathematical operations of CNNs, addressing common issues encountered by beginners learning CNNs through interactive visualization. Furthermore, interactive online scientific journals like Distill [11], [12] encourage us to move beyond traditional academic formats, making interactive articles with interactive visualizations explaining deep learning models increasingly popular.

It is noteworthy that EL-ViT is developed based on CNN Explainer. However, for comparative purposes, our visualization system offers more advanced features, with two key distinctions. Firstly, as an educational tool, the inclusion of a systematic architectural diagram of the model is essential, particularly for complex models. This allows users to grasp the entire process from input to output comprehensively. Additionally, the system incorporates explanations of terminology and source code to assist users in learning relevant concepts. Secondly, EL-ViT provides an alternative approach to enhancing the interpretability of the ViT model. This allows users not only to acquire knowledge about ViT but also to gain insights into ViT's classification decisions.

#### B. Visual Analysis for Deep Learning Model

As the field of deep learning visualization continues to evolve, in addition to serving as educational tools, there are numerous visualization systems designed to assist experts in analyzing their models and predictions, explaining how models respond to their datasets. For example, CNNVis [4] is employed to aid experts in comprehending, diagnosing, and refining deep CNN models. LSTMVis [13] and RNNVis [14] provide tools for users to explore the hidden states of RNNs. DeepNLPVis [5] effectively identifies and diagnoses issues in deep NLP models used for text classification. GNNExplainer [15] offers interpretability for predictions of any GNN-based model in any graph-based machine learning task. CorGIE [16] is designed to check whether Graph Neural Networks (GNNs) learn significant features from graphs, and GNNLens [17] facilitate the prediction error analysis of GNNs. Some tools allow users to diagnose models during training. For example, DeepEyes [18] progressively analyzes DNNs during training by sampling subregions of the input space. DGMTracker [19] enables experts to diagnose and monitor the training process of generative models by visualizing time-series data on data flow graphs. GANViz [20] offers a comprehensive understanding

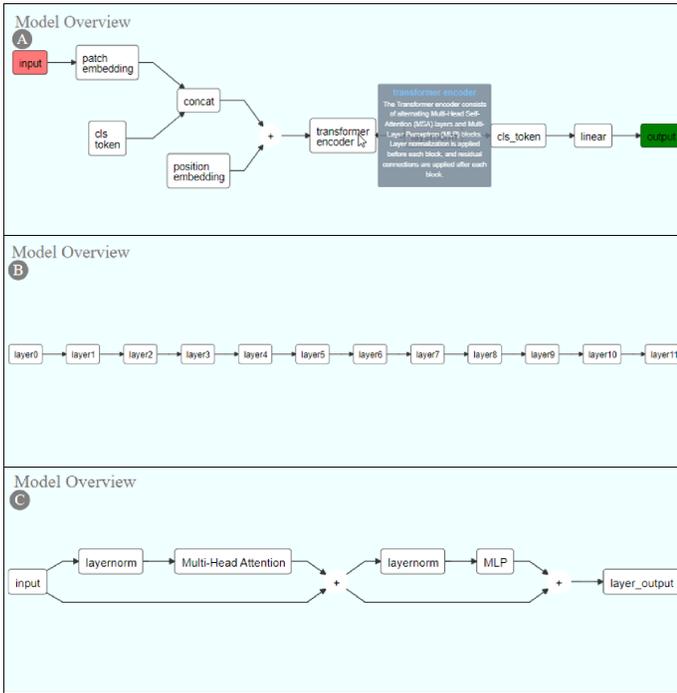


Fig. 3. Model Overview. (A) illustrates the overall structure of the ViT model. (B) showcases the 12 layers of the Transformer encoder. (C) details the composition of each Transformer block at every layer.

of the adversarial process of GAN models, helping experts assess and explain training results through multiple views.

The aforementioned work has achieved significant success in the field of deep learning visualization. Additionally, there are several interpretability methods to help us understand Transformer models. For instance, interpretability has been achieved by employing attention mechanisms to explain the model [21]–[24]. Attribution through self-attention has been used to elucidate the internal information interactions of the Transformer, aiding in the identification of critical attention heads [25]. The allocation of local correlations through deep Taylor decomposition principles, followed by the propagation of these correlations across layers, has been employed [26]. Visual explanations have been provided by assessing the similarity between two regions of images [27]. In comparison, our approach introduces a novel perspective by analyzing only the output obtained from a single image through Transformer blocks, specifically by comparing the cosine similarity between any two patches of the output.

#### IV. DESIGN GOAL

EL-VIT endeavors to create a visual platform for users to learn about and explore ViT. Nonetheless, diverse user types possess distinct demands for visualization systems. Consequently, we seek to have the visualization system accomplish the following goals:

- **Catering to both global and detailed exploration of the model simultaneously.** The ViT model possesses a deep layer structure, for example, in the case of the relatively

simpler ViT-B/16 model, it consists of three modules: the Embedding layer, Transformer Encoder, and MLP Head. In particular, the Transformer Encoder consists of 12 layers of Transformer blocks, with each block comprising LayerNorm, MLP, and multi-head attention components. Furthermore, the multi-head attention itself encompasses various types of layer structures. Hence, there is a need for a dual approach encompassing both global and detailed exploration, ensuring that users do not feel lost while navigating the system.

- **Illustrating the complete data flow process of the ViT model.** The model structure of ViT is highly intricate, with various types of layers involving distinct mathematical operations. Users often find it challenging to comprehend the transformations occurring between these layers. Given these considerations, EL-VIT aims to provide a comprehensive data flow process, aiding users in understanding how image data undergoes step-by-step transformations to yield classification predictions.
- **Achieving the interpretability of ViT.** Certain users aim to gain insight into both the practical application and the underlying reasons for the effectiveness of ViT. We propose an innovative approach that involves visualizing the cosine similarity among the patch of outputs of Transformer blocks. This visualization empowers users to independently delve into the associated content.
- **Providing a variety of interactive visualization methods to support user exploration.** To deliver an exceptional user experience, it is crucial to thoughtfully utilize diverse interactive visualization techniques. For example, when users have limited knowledge of convolutions, a combination of overview and detailed visualizations can facilitate knowledge exploration. The inclusion of animated elements can also captivate users, making it easier for them to grasp the underlying principles and implementation processes.
- **Deploying the proposed approach as a web-based application.** In order to expand the system’s accessibility to a wider audience, we have implemented the visualization tool on a web-based platform. This eliminates the necessity for any software installations, allowing users to easily access it through a web browser. This approach simplifies the process of acquiring essential knowledge about ViT and provides users with the opportunity to experience the operational procedures.

#### V. INTERFACE

The visualization system in this paper is primarily composed of four layers of visual views: Model Overview, Knowledge Background Graph, Model Detail View, and Interpretation View. An overview of the system is shown in Fig. 1. The first three layers of views explain the operational flow of ViT from the perspectives of overall structure, detailed interpretation, and mathematical operations. They enable users to comprehend the underlying mathematical operations and the transformation processes between layers in the model. The



Fig. 4. Knowledge Background Graph. (A) depicts as a force-directed graph, visualizing crucial concepts and the code’s architecture. (B) displays the specific content of the node through modal box.

fourth layer, the Interpretation View, is designed to provide in-depth understanding of the ViT model by calculating the cosine similarity between patches. Users need to click on “Interpretation View” in the top right corner of the Model Detail View.

**A. Model Overview**

When beginners first encounter a model, it’s essential to begin with an understanding of the model’s overall structure before delving into the details. Therefore, EL-ViT incorporates a Model Overview, as illustrated in Fig. 3, which assists users in constructing a mental model of the model. This aids users in gaining a better understanding of the entire process, clarifying how the input from image data ultimately leads to the output classification.

For user convenience, the model overview incorporates tooltip functionality, as depicted in Fig. 3(A). When a user hovers the mouse pointer over a specific intermediate step in the ViT model, a tooltip appears to provide a brief explanation of the term. This prevents users from feeling puzzled when encountering new terminology. Given the deep layer structure of the ViT model, representing the entire process in a single image could become cluttered. Therefore, the visualization system employs an interactive approach that combines focus+context. When a user clicks on “transformer encoder”, the interface displays its 12-layer structure. Clicking on one of the layers reveals the composition of the Transformer block, while clicking in a blank area will navigate the user to the

previous level. The specific effects are illustrated in Fig. 3(A), Fig. 3(B), and Fig. 3(C).

**B. Knowledge Background Graph**

In the process of learning about the ViT model’s operation, users must not only acquire theoretical knowledge but also understand its practical implementation. Therefore, EL-ViT incorporates a Knowledge Background Graph, primarily showcasing the concepts of relevant terms and the source code used for image classification by the ViT model. The conceptual explanations of terms assist users in understanding ViT-related content, while the source code aids users in comprehending the ViT model through its implementation process.

For example, the scenario where users encounter difficulties in comprehending the image classification implementation process of ViT. In such cases, users can refer to the Knowledge Background Graph within the visualization system. Since the model is organized by class in the source code, each node in this view corresponds to a specific class, and each edge can signify the presence of call relationships between nodes representing classes. The nodes within the view are generated using a force-directed layout algorithm. When users click on nodes of interest, a modal box appears, displaying the pytorch source code implementation process corresponding to that node. As illustrated in Fig. 4(A), if a user wishes to understand the implementation process of “ViTLayer”, they can click on the corresponding node to reveal its internal implementation details, as depicted in Fig. 4(B). This process helps users gain insights into the code and its interactions with other classes.

For the purpose of enhancing the visual quality of the graphics, and to avoid any overlap between text and graphics, in the design of force-directed graphs in this paper, text information is incorporated as nodes in the force-directed graph, participating in the computation of the force-directed layout. Additionally, text node forces are introduced, which represent attractive forces between nodes and text. The combined action of multiple forces ensures that labels and actual nodes are mutually repulsive, preventing overlap, while also exerting attractive forces to bring text and nodes closer together.

**C. Model Detail View**

In the process of learning about the ViT model, beginners frequently grapple with understanding the detailed data flow, the transformations between layers, and the specific mathematical operations within each layer. Hence, EL-ViT introduces a Model Detail View.

1) *Visualization:* Fig. 5 demonstrates the utilization of a focus+context visualization approach in the Model Detail View, systematically guiding users through each layer. This view offers insights into the internal structure of each layer and the vector transformation processes. Users can conveniently navigate backward through the layers by clicking on empty spaces, facilitating a clear understanding of the data’s transformation from the initial 224×224×3 image format to a 1×10 matrix, serving as the final classification prediction. To aid users in observing classification predictions, this paper

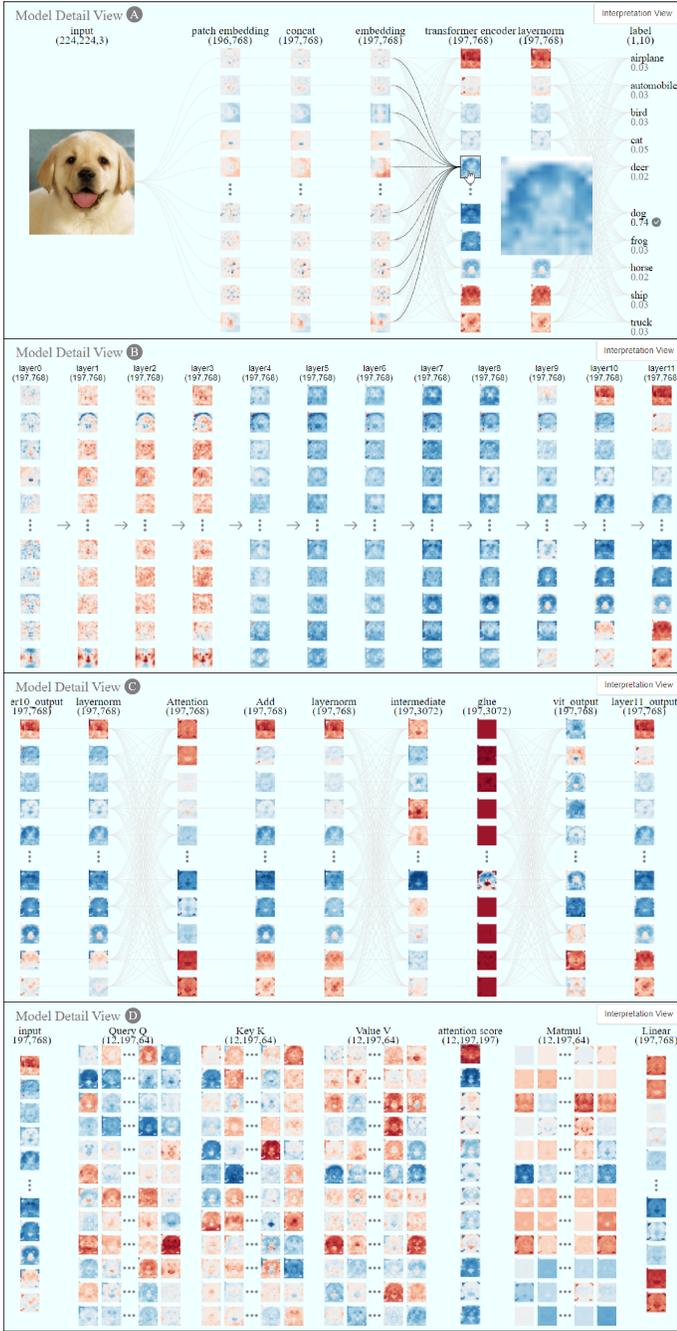


Fig. 5. Model Detail View. (A) depicts the overall model process of parameters visualization. (B) displays parameter visualizations of the outputs from 12 Transformer blocks. (C) illustrates parameter visualizations for each internal layer within a Transformer block. (D) visualizes parameters of the internal layers within the Multi-Head Attention layer.

employs iconography to highlight the class with the highest prediction probability.

To effectively visualize the graphical results, EL-VIT does not directly visualize parameters. For instance, in the embedding layer, as observed in Fig. 2, a vector of size  $197 \times 768$  is obtained through previous layer operations. This paper transposes the vector, resulting in a  $768 \times 197$  vector, which

0														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
15	16	17	18	19	20	21	22	23	24	25	26	27	28	
29	30	31	32	33	34	35	36	37	38	39	40	41	42	
43	44	45	46	47	48	49	50	51	52	53	54	55	56	
57	58	59	60	61	62	63	64	65	66	67	68	69	70	
71	72	73	74	75	76	77	78	79	80	81	82	83	84	
85	86	87	88	89	90	91	92	93	94	95	96	97	98	
99	100	101	102	103	104	105	106	107	108	109	110	111	112	
113	114	115	116	117	118	119	120	121	122	123	124	125	126	
127	128	129	130	131	132	133	134	135	136	137	138	139	140	
141	142	143	144	145	146	147	148	149	150	151	152	153	154	
155	156	157	158	159	160	161	162	163	164	165	166	167	168	
169	170	171	172	173	174	175	176	177	178	179	180	181	182	
183	184	185	186	187	188	189	190	191	192	193	194	195	196	

Fig. 6. The resulting 2D matrix reshaped from an original vector. The number within each cell indicates the original index of each element in the original vector.

can be perceived as 768 individual  $1 \times 197$  vectors. Each of these individual vectors is reshaped into a 2D matrix, where the shape of the 2D matrix and the corresponding element locations are shown in Fig. 6. Finally, the parameter values  $x_i$  are subject to transformation according to Equation 6, normalizing them within the  $[0, 1]$  range to fulfill the input criteria for color interpolation, culminating in the effect depicted in Fig. 6. In this context,  $x_i$  denotes the parameter value associated with the  $i$ -th patch, range denotes the difference between the maximum and minimum values among these 197 patches, and min represents the minimum value among these 197 patches.

$$x_i = \frac{x_i - \min}{\max - \min} \quad (6)$$

2) *Interactive View*: Users need to comprehend not only the overall structure of the model but also the underlying principles and mathematical operations of each layer. Therefore, this paper presents the internal workings of each layer in the form of modal boxes to enhance the interactive exploration experience for users.

For the visualization of convolution operations, as illustrated in Fig. 7(A), the input image is  $224 \times 224 \times 3$ , divided into 196 patches, with each patch sized at  $16 \times 16 \times 3$ . Each patch is subjected to convolution using a  $16 \times 16 \times 3$  kernel and is subsequently augmented with bias terms, yielding a final  $14 \times 14$  matrix. EL-VIT utilizes animation to illustrate the complete convolution procedure, demarcating the currently convolving patch with a rectangular frame for user clarity.

For the visualization of the non-linear activation function gelu, as illustrated in Fig. 7(B), the view initially provides a visualization of the gelu function. When the input value is greater than 0, the gelu function essentially produces linear output, whereas when the input value is less than 0, the gelu function generates non-linear but continuous output. This view also employs animation to visualize details, with textual information accompanying the movement of the rectangular box, thus aiding users in observing data changes.

For the visualization of the embedding layer, as illustrated in Fig. 2, the embedding layer is a result of the concat layer combined with positional embedding, as demonstrated

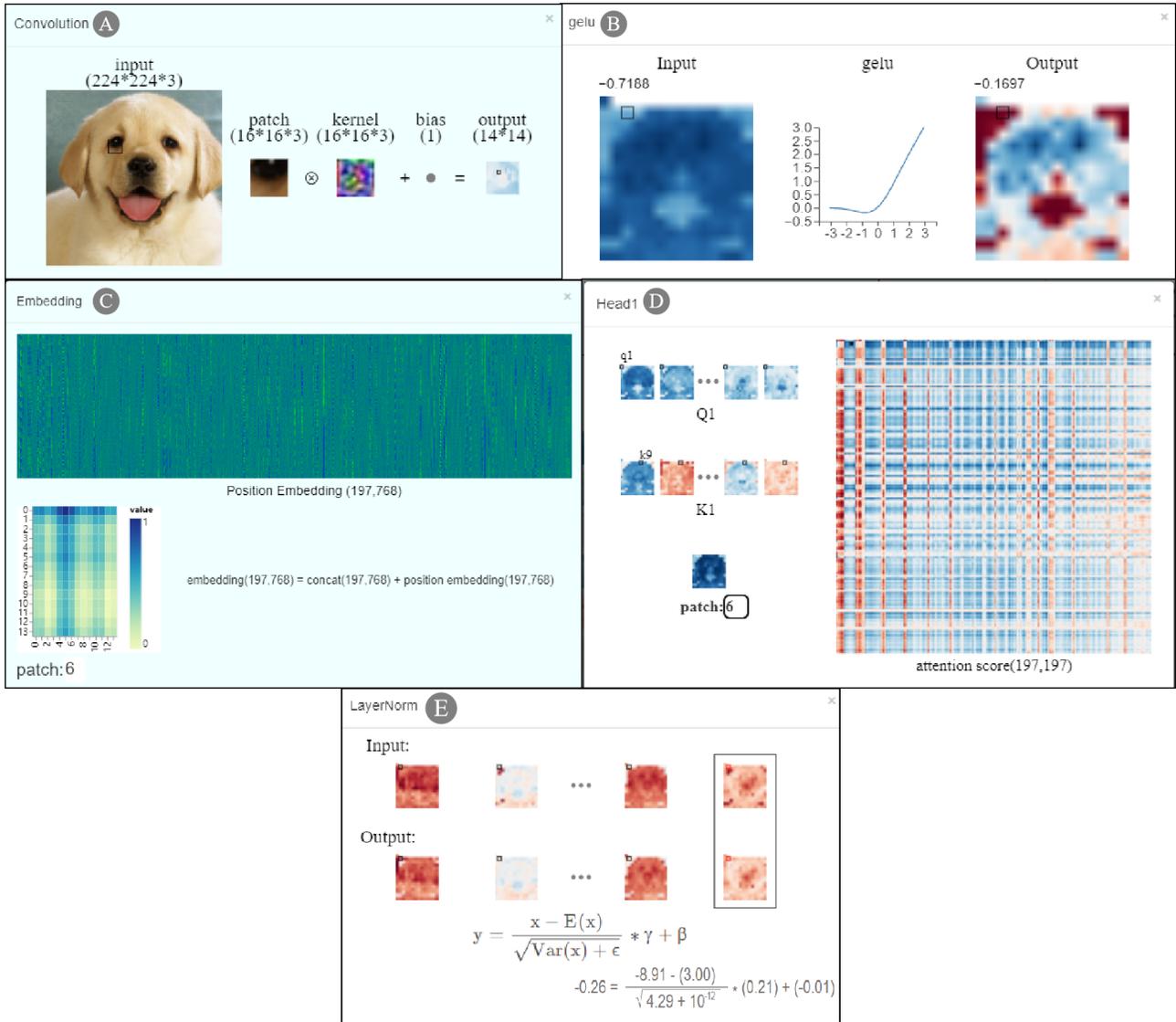


Fig. 7. Interactive Views. (A) illustrates the convolution process. (B) presents the graph of the gelu function along with its input and output results. (C) depicts the embedding process while visualizing positional information. (D) shows the computation process of attention scores in the first head. (E) displays the operation of LayerNorm.

in Fig. 7(C). Initially, the magnitudes of vector values are visually represented through a heatmap. To facilitate users in comprehending positional embedding more effectively, this visualization method employs a heatmap to illustrate the similarities in size between a particular patch and others. For instance, inputting “6” in the text box will reveal the positional information for the sixth patch. Upon observation, users will notice that the colors are deeper in the row and column corresponding to this patch, aligning with their understanding of positional information.

For the visualization of attention scores, as illustrated in Fig. 7(D), attention scores are presented in the form of a heatmap, positioned to the right of the modal box. Animated sequences are employed to guide users in comprehending the computation between Q and K for an individual head, thereby

providing insight into the origin of the attention score matrix. This view also incorporates an exploratory feature designed to assist users in extracting specific patch information for the current head, enabling users to understand the knowledge acquired by the current patch during the forward process. For example, when the number “6” is input into the text box, it reveals the weight information associated with the vector of the 6th patch and presents it visually in image format.

For the visualization of LayerNorm, which normalizes different features for each patch to ensure a stable distribution, as illustrated in Fig. 7(E), the black small square outlines the region representing the normalization of the current patch’s 768-dimensional features. It calculates the mean and variance of the current distribution, where  $\gamma$  and  $\beta$  denote the standard deviation and mean after normalization. Due to

the high dimensionality, only a subset of features can be visually displayed in this paper. The red square frames the region representing the normalization of a specific dimension within the current patch, aligning textual information with the corresponding computational formula and moving along with the red frame. To further emphasize the visualization effect, this paper employs a large rectangular box to highlight the dimensions being computed. This view is presented through animation, aiming to engage users in the learning process.

3) *Text Detail*: In the case of certain basic operations, EL-ViT opts not to employ interactive views for explanations but rather employs simple text annotations for visualization, as illustrated in Fig. 8. The image on the right is obtained by adding a CLS token to the left one. Users can simply hover their mouse over the image to access text annotations for explanation. Additionally, bounding boxes within the image highlight the position where CLS is added. Throughout this visualization system, several similar annotations exist to assist users in exploring information.

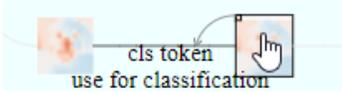


Fig. 8. Text annotation visualization.

#### D. Interpretation View

EL-ViT adopts a novel perspective for model interpretation. For the Transformer encoder, it consists of 12 layers of Transformer blocks, and our focus lies on the output of these 12 Transformer blocks. As depicted in Fig. 5(B), the output of any layer in the Transformer block is a vector of size  $197 \times 768$ . We assume that  $\beta \in R^{768}$  represents an arbitrary token, such that the cosine similarity between any two tokens  $\beta_i$  and  $\beta_j$  is given by:

$$s(\beta^i, \beta^j) = \frac{\beta^i \cdot \beta^j}{\|\beta^i\| \cdot \|\beta^j\|} \quad (7)$$

For the original  $197 \times 768$  vectors, this paper divides them into two parts: the CLS token and the remaining  $196 \times 768$  vectors. Subsequently, cosine similarity calculations are performed between the CLS token and the other 196 tokens to obtain a  $1 \times 196$  vector. Finally, this vector is reshaped into a  $14 \times 14$  image. The specific results are shown in Fig. 9. The heatmap representation visually conveys the magnitude of cosine similarity values, with bluer hues signifying greater similarity.

#### E. Web-based Implement

EL-ViT is a web-based implementation that does not involve any backend programming languages. We have employed TensorFlow.js [28] in conjunction with d3.js [29] to realize the visualization of the model. Users can experience ViT simply by accessing the webpage on the frontend. The model used in this paper [30] has been imported from Hugging

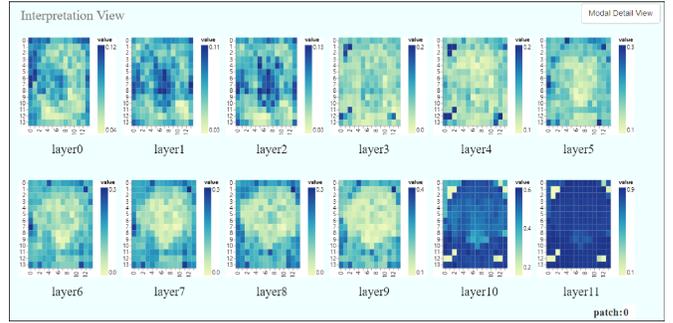


Fig. 9. Interpretation View. This view allows for the alteration of text box values, facilitating the observation of the similarity relationship between a particular patch and the remaining 196 patches.

Face and is a fine-tuned ViT model on the cifar10 dataset, employed for image classification.

## VI. USAGE SCENARIOS

### A. Learning the Forward Process

When computer vision students, who are learning about image classification, discover that the classification performance of ViT is superior to CNN, they decide to enhance the accuracy of image classification by learning and applying ViT since they are not very familiar with the ViT model. Consequently, they embark on exploring EL-ViT.

Firstly, the students begin their exploration of the visualization system by starting with the model overview, as shown in Fig. 3. This initial step provides them with a general understanding of the entire operational process of ViT, helping them establish a mental model. Subsequently, the students seek to delve deeper into the details, encountering difficulties while exploring the model detail view. They are uncertain about why a  $224 \times 224 \times 3$  image data is transformed into  $196 \times 768$  after patch embedding. By clicking on the corresponding image to reveal a modal box, as illustrated in Fig. 7(A), they swiftly comprehend that this is a convolutional process. Through animations and their foundational knowledge, they efficiently calculate the output dimensions to be precisely  $196 \times 768$ , aligning with the intended content of the visualization. Continuing their exploration through the subsequent layers, the students ultimately gain insight into the step-by-step calculation of classification predictions. Due to the model's complexity, the students face challenges during code replication. Consequently, they delve into the knowledge background graph to understand the code's construction, as shown in Fig. 4. For instance, if a user aims to create "ViT-Model", it consists of components such as "ViTEncoder" and "ViTEmbedding", with each layer housing various sub-layers. The students find that this interactive tool makes exploring ViT more engaging than traditional learning methods like lectures or video tutorials.

### B. Interpreting the Workings

In terms of Transformer interpretability, in most cases, users analyzed attention weights. Take the attention scores in

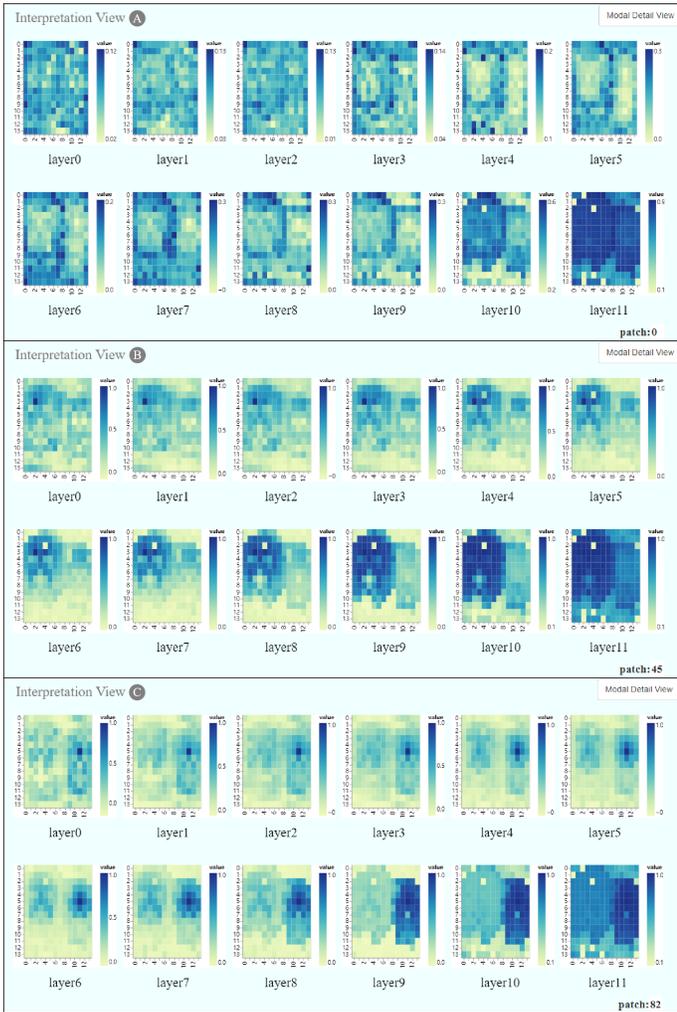


Fig. 10. Similarity view of patch. (A) illustrates the similarity map between the first patch (CLS) and the 196 patches related to the image. (B) displays the similarity map between the 45th patch (patch corresponding to the dog) and the 196 patches related to the image. (C) shows the similarity map between the 82nd patch (patch corresponding to the cat) and the 196 patches related to the image.

Fig. 7(D) as an example. For a  $197 \times 197$  vector, this paper extracts the CLS (0th dimension)  $1 \times 197$  vector, then reshapes the subsequent 196 parameters into a  $14 \times 14$  image. However, this approach may not always be effective. Therefore, users can activate the interpretation view, as illustrated in Fig. 9. For a single entity image, users can clearly see that the patch corresponding to the dog has relatively dark colors, indicating that CLS is very similar to the patch corresponding to the dog and dissimilar to the background. For multiple entities, as shown in Fig. 10(A), CLS is similar to the patches corresponding to both the cat and the dog, resulting in high probabilities for both cat and dog outputs.

As shown in Fig. 10(B), when we changed the patch in the lower right corner to 45, corresponding to the dog, all patches associated with the dog become quite similar. Similarly, when we replaced the patch with the one corresponding to the cat,

there was a corresponding effect, as shown in Fig. 10(C). This observation aligns well with the findings of Nguyen et al [31]. We believed that it was precisely the spatial similarity among patches of the same object that leads to dimension reduction and clustering effects. Moreover, by closely examining the data, we hypothesized that shallow-layer patches interact with the surrounding patches, resulting in high similarity among patches corresponding to the cat in the vicinity. It is only in the deeper layers that all patches corresponding to the cat become similar.

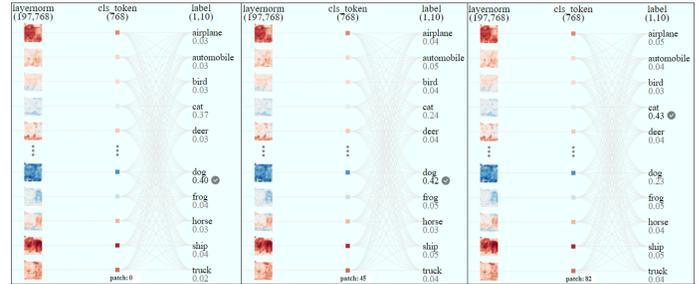


Fig. 11. The left, middle, and right figures represent the calculated prediction probability maps for the 0th patch (CLS), the 45th patch (corresponding to the dog), and the 82nd patch (associated with the cat), respectively.

To further explore, we opened the model detail view and clicked on the label tags to reveal hidden views, as shown in Fig. 11. The left, middle, and right panels represent the probabilities of various classes computed for the 0th patch (CLS), the 45th patch (patch corresponding to the dog object), and the 82nd patch (patch corresponding to the cat object), respectively. We observed that when using the patch associated with the dog object instead of CLS for the final classification prediction, the probability of predicting as a dog increased, while the probability of predicting as a cat decreased. Similarly, when using the patch corresponding to the cat object instead of CLS for the final classification prediction, the probability of predicting as a cat increased, while the probability of predicting as a dog decreased.

## VII. LIMITATIONS AND FUTURE WORK

**Lacking of training process and backpropagation.** In the process of learning deep learning models, backpropagation also plays a vital role. However, in EL-ViT, all the parameters are imported, and the lack of a backpropagation process makes it challenging for beginners to understand how these parameters are derived. In the future, we plan to enrich the content of EL-ViT and gradually visualize the backpropagation process.

**Extending visualization form.** Due to the high dimensionality and large number of parameters in ViT, it is challenging to visualize all the details adequately. To address this, we aim to extend our visualization methods in the future by incorporating greater interactivity. This approach will better assist users in exploring and gaining detailed insights into specific aspects of interest.

**Further exploring the interpretability of the model.** While we have explored the interpretability of the model from

another perspective, whether through the model’s outputs or by investigating its interpretability through attention weights, it is essential to recognize that neither approach can provide a complete explanation of the model. Both of these methods only represent a small portion of the entire model. We hope that in the future, further research will enable us to comprehensively understand these black-box models.

**Evaluation of the educational effectiveness.** EL-VIT provides interpretability explanations and serves as an educational tool, aiding both experts and novices in knowledge acquisition. Future plans include incorporating surveys to further explore the learning experiences of beginners.

### VIII. CONCLUSION

In this paper, we introduce a user-friendly interactive visualization tool that employs intuitive visualization techniques to assist users in exploring ViT models. It caters to both experts and beginners, facilitating the acquisition of necessary knowledge. By employing cosine similarity on the output of each layer of the Transformer, we observe that patches corresponding to the same object exhibit higher similarity, and the CLS token also shares high similarity with patches corresponding to objects. Finally, we demonstrate the effectiveness of the visualization system through two usage scenarios.

### ACKNOWLEDGMENT

We thank anonymous reviewers for their valuable comments. This research was supported by Lee Kong Chian Fellowship awarded to Dr. Yong Wang by SMU. The demonstration video for this visualization system is located at <https://youtu.be/MUmk0Fi0JKk>.

### REFERENCES

- [1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
- [2] Wang, Z.J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M., Chau, D.H.P.: Cnn explainer: learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics* 27(2), 1396–1406 (2020)
- [3] Wang, Q., Yuan, J., Chen, S., Su, H., Qu, H., Liu, S.: Visual genealogy of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics* 26(11), 3340–3352 (2019)
- [4] Liu, M., Shi, J., Li, Z., Li, C., Zhu, J., Liu, S.: Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics* 23(1), 91–100 (2016)
- [5] Li, Z., Wang, X., Yang, W., Wu, J., Zhang, Z., Liu, Z., Sun, M., Zhang, H., Liu, S.: A unified understanding of deep nlp models for text classification. *IEEE Transactions on Visualization and Computer Graphics* 28(12), 4980–4994 (2022)
- [6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* 30 (2017)
- [7] Karpathy, A.: Convnetjs mnist demo. <https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html>.
- [8] Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., Chen, A.: Teachable machine: Approachable web-based tool for exploring machine learning classification. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–8 (2020)
- [9] Smilkov, D., Carter, S., Sculley, D., Viégas, F.B., Wattenberg, M.: Direct-manipulation visualization of deep networks. *arXiv preprint arXiv:1708.03788* (2017)
- [10] Kahng, M., Thorat, N., Chau, D.H., Viégas, F.B., Wattenberg, M.: Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics* 25(1), 310–320 (2018)
- [11] Sanchez-Lengeling, B., Reif, E., Pearce, A., Wiltchko, A.B.: A gentle introduction to graph neural networks. *Distill* 6(9), 33 (2021)
- [12] Daigavane, A., Ravindran, B., Aggarwal, G.: Understanding convolutions on graphs. *Distill* 6(9), 32 (2021)
- [13] Strobelt, H., Gehrmann, S., Pfister, H., Rush, A.M.: Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics* 24(1), 667–676 (2017)
- [14] Ming, Y., Cao, S., Zhang, R., Li, Z., Chen, Y., Song, Y., Qu, H.: Understanding hidden memories of recurrent neural networks. In: *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24 (2017). IEEE
- [15] Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019)
- [16] Liu, Z., Wang, Y., Bernard, J., Munzner, T.: Visualizing graph neural networks with corgie: Corresponding a graph to its embedding. *IEEE Transactions on Visualization and Computer Graphics* 28(6), 2500–2516 (2022)
- [17] Jin, Zhihua, Yong Wang, Qianwen Wang, Yao Ming, Tengfei Ma, and Huamin Qu. "Gnnlens: A visual analytics approach for prediction error diagnosis of graph neural networks." *IEEE Transactions on Visualization and Computer Graphics* (2022).
- [18] Pezzotti, N., H<sup>o</sup>llt, T., Van Gemert, J., Lelieveldt, B.P., Eisemann, E., Vilanova, A.: Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE Transactions on Visualization and Computer Graphics* 24(1), 98–108 (2017)
- [19] Liu, M., Shi, J., Cao, K., Zhu, J., Liu, S.: Analyzing the training processes of deep generative models. *IEEE Transactions on Visualization and Computer Graphics* 24(1), 77–87 (2017)
- [20] Wang, J., Gou, L., Yang, H., Shen, H.-W.: Ganviz: A visual analytics approach to understand the adversarial game. *IEEE Transactions on Visualization and Computer Graphics* 24(6), 1905–1917 (2018)
- [21] DeRose, J.F., Wang, J., Berger, M.: Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics* 27(2), 1160–1170 (2020)
- [22] Strobelt, H., Gehrmann, S., Behrisch, M., Perer, A., Pfister, H., Rush, A.M.: Seq2seq-viz: A visual debugging tool for sequence-to-sequence models. *IEEE Transactions on Visualization and Computer Graphics* 25(1), 353–363 (2018)
- [23] Vig, J.: A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714* (2019)
- [24] Wang, Z.J., Turko, R., Chau, D.H.: Dodrio: Exploring transformer models with interactive visualization. *arXiv preprint arXiv:2103.14625* (2021)
- [25] Hao, Y., Dong, L., Wei, F., Xu, K.: Self-attention attribution: Interpreting 19 information interactions inside transformer 35(14), 12963–12971 (2021)
- [26] Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791 (2021)
- [27] Black, S., Stylianou, A., Pless, R., Souvenir, R.: Visualizing paired image similarity in transformer networks. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3164–3173 (2022)
- [28] Smilkov, D., Thorat, N., Assogba, Y., Nicholson, C., Kreeger, N., Yu, P., Cai, S., Nielsen, E., Soegel, D., Bileschi, S., et al.: Tensorflow.js: Machine learning for the web and beyond. *Proceedings of Machine Learning and Systems*, 1, 309–321 (2019)
- [29] Bostock, M., Ogievetsky, V., Heer, J.: D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 17(12), 2301–2309 (2011)
- [30] <https://huggingface.co/aaraki/vit-base-patch16-224-in21k-finetuned-cifar10/discussions>
- [31] Nguyen, V.-A., Dinh, K.P., Vuong, L.T., Do, T.-T., Tran, Q.H., Phung, D., Le, T.: Vision transformer visualization: What neurons tell and how neurons behave? *arXiv preprint arXiv:2210.07646* (2022)