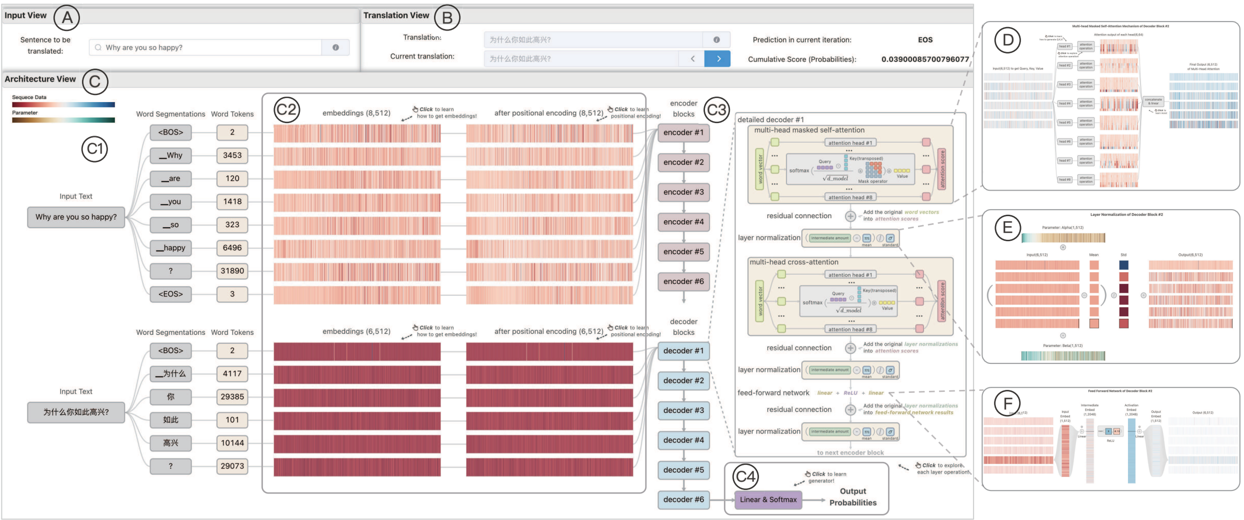
TransforLearn: Interactive Visual Tutorial for the Transformer Model

Lin Gao, Zekai Shao, Ziqin Luo, Haibo Hu, Cagatay Turkay and Siming Chen



Input View (A)

Sentence to be translated: Why are you so happy?

Translation View (B)

Translation: 为什么你如此高兴?

Current translation: 为什么你如此高兴?

Prediction in current iteration: EOS

Cumulative Score (Probabilities): 0.03900085700796077

Fig. 1: With TransforLearn, learners can gain an understanding of the Transformer structure and the process of machine translation. Input view **(A)** provides an interface for the text to be translated. Translation view **(B)** displays the model's translation results and current translation progress, helping users in task-driven exploration. Architecture view **(C)** provides an overview of the visualized model structure and data flow, with sub-views **(C1-C4)** that support the close exploration of computational processes. Once enabled, the detailed view **(C3)** displays the attention mechanism view **(D)**, layer normalization view **(E)**, and feed-forward network view **(F)**. These views not only show the data flow and operational details but also support multiple interactions.

Abstract—The widespread adoption of Transformers in deep learning, serving as the core framework for numerous large-scale language models, has sparked significant interest in understanding their underlying mechanisms. However, beginners face difficulties in comprehending and learning Transformers due to its complex structure and abstract data representation. We present TransforLearn, the first interactive visual tutorial designed for deep learning beginners and non-experts to comprehensively learn about Transformers. TransforLearn supports interactions for architecture-driven exploration and task-driven exploration, providing insight into different levels of model details and their working processes. It accommodates interactive views of each layer's operation and mathematical formula, helping users to understand the data flow of long text sequences. By altering the current decoder-based recursive prediction results and combining the downstream task abstractions, users can deeply explore model processes. Our user study revealed that the interactions of TransforLearn are positively received. We observe that TransforLearn facilitates users' accomplishment of study tasks and a grasp of key concepts in Transformer effectively.

Index Terms—Deep learning, Transformer, Visual tutorial, Explorable explanations

1 Introduction

Deep learning models are now commonplace in various industries and are being applied to increasingly more complex problems. Transformer, a kind of deep learning model, has become one of the hot spots in research and has become the preferred method for many tasks. For example, to generate a piece of poetry, the Generative Pre-trained Transformer (GPT) [37] is used to solve the text generation problem, and to classify images, the Vision Transformer (ViT) [13] is used. Due to such wide applicability, there has been an immense interest in learning about deep learning. However, these deep learning models often involve complex structures and esoteric mathematical formulas, which create barriers for beginners. In recent years, visual and interactive methods have shown to be effective in explaining the working mechanisms and concepts of complex models [29,43,57] for seasoned model builders. This is encouraging for interactive visual approaches targeting broader audiences and this paper takes steps towards this goal.

Transformer has emerged as a prominent and widely-used tool in natural language processing (NLP) due to its exceptional performance, first proposed by Google in 2017 [45] to tackle neural machine translation tasks. Transformer effectively resolves the issues of slow processing speeds and sequence length limitation that traditional neural network models, such as Recurrent Neural Networks (RNNs) [42] and Convolutional Neural Networks (CNNs) [35]. Due to their capabilities in handling long sequence data, comprehending the changes within Transformers is difficult. Meanwhile, it is particularly challenging to understand the intricate operations of modules and transformations of data dimensions for beginners. To better support learning, beginners

- *Lin Gao, Zekai Shao, Ziqin Luo and Siming Chen are with School of Data Science, Fudan University. S. Chen is also with Shanghai Key Laboratory of Data Science. S. Chen is the corresponding author. E-mail: simingchen@fudan.edu.cn.*
- *Haibo Hu is with Chongqing University. E-mail: haibo.hu@cqu.edu.cn. H. Hu is the co-corresponding author.*
- *Cagatay Turkey is with the University of Warwick. E-mail: Cagatay.Turkay@warwick.ac.uk.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

should have more extensive and active participation with Transformer. Most existing Transformer tutorial methods, however, rely on graphics and text for sequential narration rather than free exploration, such as the examples from Peter Bloem [6] and Samira Abnar [1]. These methods fall short of offering an immersive learning experience for novices. Consequently, there is a pressing need for comprehensive, interactive, and visually engaging tutorial tools tailored for Transformers to bridge this educational gap.

Contribution. We propose TransforLearn for beginners as a tutorial tool for Transformers. TransforLearn uses a visual approach to provide learners with a better learning experience through interactive exploration. Our major contributions can be listed as:

- **TransforLearn, the first interactive visual explanation system as a tutorial for Transformers.** TransforLearn provides a hierarchical overview of the model architecture. It combines interactive displays of data flow transformations and mathematical formulas, seamlessly integrating the model's top-level architecture with downstream numerical features. TransforLearn aids users in gaining a comprehensive understanding of the model architecture and its intricate execution processes.
- **Novel interactive exploration approaches to facilitate training on Transformer models.** We support **architecture-driven exploration** guided by the structure and **task-driven exploration** based on the iteration of downstream tasks. These distinct interactive modes are designed to assist beginners in comprehending the intricacies of Transformer.
- **Evaluating the effectiveness of our work.** A user study confirms that TransforLearn provides users with an immersive learning experience. After using the system, users generally exhibited better performance when completing Transformer-related tasks.

With TransforLearn, we contribute to the growing literature on employing interactive visualization techniques to improve the interpretability [14, 16] of AI methods and establish a stronger role for visualization in enhancing the AI literacy [30] of broader audiences.

2 Background on Transformers

This section offers an introduction to Transformer, establishing the foundation for our work. With the rapid development of NLP, Transformers serve as key kernels supporting the most popular large language models. OpenAI introduced the GPT in 2018 [37], which exclusively utilizes the Transformer decoder for language modeling and generation. During the same period, Google introduced BERT [12], a groundbreaking model reliant solely on the Transformer encoder for pre-training and fine-tuning. Additionally, various enhanced or extended versions, including XLNet [56] and GPT-2/3 [8, 38], as well as Instruct GPT [36], have further enriched the landscape.

Structurally, the Transformer belongs to the encoder-decoder design. The encoder block features a multi-head self-attention mechanism and a positional feed-forward network, while the decoder block includes a multi-head cross-attention mechanism. These elements are linked by a combination of two operations: residual connection [17] and layer normalization [28]. In machine translation tasks, the input text is expressed by embedding and positional encoding to obtain numerical representations of word embedding.

3 Related Work

We reviewed related work about visual interpretation for deep learning models, with a specific focus on the Transformer model. We also looked at existing visual tutorial tools for deep learning models.

3.1 Visualization for understanding deep learning models

The interpretability and transparency of deep learning models remain a persistent issue [4]. An increasing number of researchers and practitioners are actively exploring methods to comprehend, compare, and enhance deep learning models [32, 49]. In recent years, the contribution of visualization to deep learning interpretability has garnered widespread acclaim [40, 59].

The most direct purpose of understanding deep learning models is to know how the models make decisions and what they learn. For example, LSTMVis [44] uses parallel coordinate plots to visually interpret the hidden features of long short-term memory networks. CNNVis [28] represents a deep CNN as a directed acyclic graph and use hybrid visualization techniques to reveal data streams as well as interactions between neurons. Recent efforts have also been concentrated on unified interpretations of specific domains, with DeepNLPVis [25] using a unified information-based measure [15] for NLP models, and M2Lens [50] for multimodal sentiment analysis.

Another key use of interactive visualization systems in model interpretation is model improvement and debugging. In order to improve the training efficiency, DeepTracker [27] combines a hierarchical indexing mechanism and novel cube-style visualizations to explore CNN training logs. DQNViz [48] demonstrates the training details of deep Q networks and conducts comprehensive analysis in the surrogate model's experience space. Wongsuphasawat et al. [53] use data flow diagrams to express the calculation process of machine learning algorithms and support users to construct algorithms independently. Meanwhile, for optimizing data sets, ShortcutLens [20] offers insights into the coverage and semantic understanding of instances with Shortcuts problems [31]. Hohman et al. [18] outlined three common target users for visual analytics in deep learning: model developers, model users, and non-experts. However, most above systems focus on providing new perspectives and methods for experts and are not suitable as tutorial education tools.

3.2 Visual interpretation of Transformers

Transformer [45] has become a favorite in NLP [58], resulting in a variety of derived models [26]. Significant efforts have been put into visualizing these architectures, predominantly focusing on interpretation and tutorial-based explanations [7].

Interpretation emphasizes embedding and attention mechanisms through both static visualizations [10] and interactive tools [46, 47]. Attviz [62] is a representative online tool that explores the correlation between self-attention scores and real data, illustrating attention matrix statistics from sequence and overall perspectives. Attention flows [11] adopts a ring-shaped design structure to support the mining of attention weights within layers, between layers, and between attention heads. Dodrio [51] summarizes the role of different attention heads, focusing on the influence of attention weights in syntactic structure and semantic information. Recently, Shao et al. [41] propose VEQA to explore the decision flow of a complex transformer-based model for the open-domain question-answering task. However, most of the work mentioned above is aimed at model developers or experts and focuses on model analysis, which is not suitable for beginners.

Several tutorial visualizations are developed recently. Jay Alammar's blogs [2, 3] analyze how the BERT model works, serving as introductory tutorials. Combined with the Transformer principle [45] and the interpretation of the implementation code, Harvard NLP provides an online tutorial [33]. Video introductions of models, such as the highly viewed *Transformer Neural Networks* [9] on YouTube, are becoming increasingly popular. Although the above popular tutorials clearly describe the structure and working mechanism, they lack interaction and exploration with the actual data flow or task, which is a gap that our work aims to fill.

3.3 Visual tutorial tools for deep learning models

The popularity of deep learning must have attracted the attention of numerous non-experts. Visual tutorial tools serve as the main way to develop the intuition about hyper-parameter changes and structural adjustments of deep learning models for non-experts. Early visual tutorial tools were mostly text-based, with visuals aiding key concept explanations. For example, Chris Olah's series of interactive blogs [34] combine text descriptions with graphical diagrams to explain many fundamental concepts and mathematics in deep learning.

To link unfamiliar layer operations with complex model structures, interactive visualization tools have emerged in later work. Typically, TensorFlow Playground [43] allows users to dynamically adjust settings such as the number of hidden layers, the number of neurons, and activation functions in Multilayer Perceptron (MLP), which fully demonstrates the advantages of interactive visualization tools. GAN Lab [23] integrates model architecture, sample distribution, and predictive performance into a system to examine multiple aspects of GAN performances. Wang et al. [52] proposed that it is very challenging for beginners to organically associate the underlying mathematical knowledge with the top-level model structure. In response to this difficulty, they propose CNN Explainer to help users understand the dynamic changes of data flow in the underlying components. To our best knowledge, our work is the first interactive visual tutorial for Transformer.

4 Preliminary study for Requirement analysis

We report the results from the interviews we conducted with domain experts and student groups to extract the challenges encountered during the learning process and compile them as design requirements.

4.1 Interviews and surveys

To understand learning challenges and inform our designs, we interviewed experienced deep-learning lecturers at a university and conducted a survey with their students.

Initially, we gathered feedback from two university lecturers (T1 and T2) about teaching methods and desired improvements, who both teach NLP and emphasize Transformers in their curriculum. In 30-minute interviews, both revealed a common teaching strategy: manually break down Transformer into multiple steps and discuss them in a sequence of slides. They also expressed their desired features for a learning tool: (1) enabling hands-on experience to enhance students' understanding, (2) developing interactive resources that permit the exploration of both the overall Transformer architecture and its individual components, and (3) clarifying data transformations and making complex mathematical operations more engaging for beginners.

We also recruited students who had studied Transformers to participate in an online survey. We received 25 (7 female, 18 male) responses. This group included 10 Ph.D. students, 10 M.S. students, and 5 undergraduates, all from Computer Science and Data Science disciplines. The survey asked about the key challenges in learning and applying Transformers from various aspects, and what features would be helpful in an interactive tool for beginners. We offered options based on the lecturers' feedback but also allowed students to add their own opinions. The summary of their responses can be found in Fig. 2.

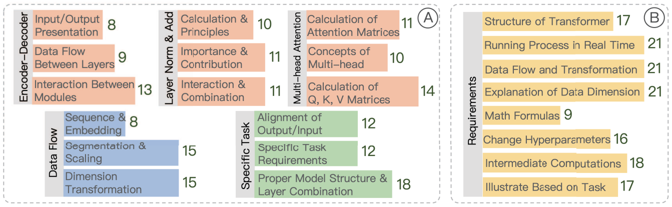


Fig. 2: The difficulties of learning Transformers (A) and the requirements for the visual tutorial design (B) resulting from the student surveys. Different color histograms map different aspects, among which red is about model structure, blue is about data flow, green is the difficulties in specific tasks, and yellow is the user requirements. The number next to the histogram indicates the number of students who selected the option.

4.2 Challenges for learning Transformer

Through interviews with lecturers and student surveys, we identified challenges in learning Transformers. In the following, we distill the fundamental learning needs of students in learning this concept.

C1 **Complex model structures with multiple layer operations.** T1 suggests that traditional teaching methods such as PowerPoint, which emphasizes theoretical knowledge, can oversimplify and neglect contextual details when explaining complex model structures and multi-layer operations. Moreover, students reportedly struggle to grasp the intricacies of certain aspects of the Transformer structure (Fig. 2A), such as the Encoder-Decoder interaction and attention matrix calculations.

C2 **Data flow and transformation.** T2 observed students' reluctance to learn mathematical formulas and limited grasp of long sequence representations in the data flow. Feedback from student surveys mirrored this (Fig. 2A), indicating a partial grasp of the data transformation and segmentation within learning materials.

C3 **The gap between grasping model structure and its practical use in downstream tasks.** According to surveys, after completing a course on Transformers or utilizing online learning materials, most beginners find themselves perplexed when attempting to apply the learned concepts to their own projects. T1 found that several of these learning materials tend to emphasize theory more than practical implementation.

C4 **The necessity of guidance and feedback.** Survey results show that students commonly use traditional Transformer tutorials, such as blogs and videos. Yet, T2 highlights their overwhelming information density and lack of interactivity, posing challenges for effective learning, particularly for beginners.

4.3 Design goals for TransforLearn

Based on the learning challenges identified (Sec. 4.2) and desired features illustrated in student surveys (Fig. 2B), we present the following design objectives for TransforLearn. Our target users are beginners with a foundational understanding of deep learning who are eager to gain a comprehensive understanding of Transformers. Additionally, we envision TransforLearn serving as a valuable teaching aid during lectures.

G1 **A visual summary of the model architecture and data flow.** TransforLearn needs to demonstrate the implementation and module interaction from the high-level model architecture to the underlying mathematical mechanism (C1). As suggested by T1, the system must ensure coherence in presenting contextual details to aid beginners in navigating the modules without becoming disoriented.

G2 **An interactive interface for layer operations and mathematical formulas.** For beginners, it is important to understand layer operations in detail and grasp the data flow in real-time. Therefore, we need to provide interactive interfaces for each level of operations and mathematical formulas, allowing users to understand how the model works (C1) and gain insights into data transformations in different dimensions (C2).

G3 **Exploration mode between module levels based on downstream tasks.** To more effectively align the abstract model with specific tasks (C3), it is necessary to provide an interactive mode that empowers users to acquire fresh perspectives on the model's structure and data manipulations, based on their engagement with downstream tasks.

G4 **Self-directed and immersive learning experiences.** To overcome beginners' reluctance to mathematical formulas and prevent them from getting lost in these details (C4), we need a logically connected and visually guided tutorial that enables users to actively explore the detailed module of Transformers.

5 TransforLearn: Interactive Visual Tutorial for the Transformer Model

The implementation of TransforLearn is built on the foundational Transformer model [45]. We visualize the forward propagation process of the training model: converting an input text to be translated into translation results. The workflow of TransforLearn is shown in Fig. 3. As illustrated in Sec. 4.1, to better assist beginners in overcoming the hurdle of aligning model input and output with task requirements, we propose two exploration modes: **architecture-driven exploration** and **task-driven exploration**. Users can grasp the overall architecture and data flow (G1) by architecture-driven exploration

(Sec. 5.1). Based on the knowledge of the architecture overview, users can explore the inner layers and study the formula parameters through certain interactions (G2). Based on task-driven exploration (Sec. 5.2), users can change the decoding time step in real time to derive further insights (G3, G4). In Sec. 5.3, we discuss the interaction between the two modes.

Color schemes. In our visualization of the real data flow, we map red to lower values and blue to higher values. The corresponding color scheme for parameter information is mapped from yellow to green, with yellow indicating lower numerical values and green indicating higher numerical values. In order to provide smooth transitions between the colors, we use white as a neutral color. All of these color schemes are presented in the form of a heat map.

5.1 Architecture-driven Exploration

The system presents an overview of the Transformer architecture and detailed modules during architecture-driven exploration (Fig. 3). In this section, we will simulate the user's exploration approach to introduce views. The system supports drill-down when the model components contain multiple operations.

5.1.1 Input: Change input text

The input view displays the model's input text. Users can input the English text they want to be translated within the designated input box and submit it to the system back-end by pressing Enter.

5.1.2 Tokenize: Divide the text into tokens

The tokenize view (Fig. 1C1) shows the process of converting text into a sequence of word elements, consisting of word segmentation and token. The view employs a horizontal tree diagram structure to visually depict the process of text segmentation, with each word splitting corresponding to the token individually. Hovering will trigger the text description of the relevant module to help beginners have a more intuitive understanding of the concept.

5.1.3 Embedding: Generate word embeddings

After tokenization, word tokens are indexed to the word embeddings in Fig. 1C2. TransforLearn supports the drill-down by clicking on the heat matrix to explore the correspondence of indexing and the computational process of positional encoding (G2).

Explaining the correspondence of indexing. Each word element corresponds to an embedding vector, and multiple word element embedding vectors form an embedding matrix. We visualize the process of obtaining the embedding representation of a word element by mapping the index values to the relative positions of the embedding matrix. Users can view the corresponding relationship by hovering. Also, we use a textual description of the shape of each matrix and use colors to characterize each element value in the embedding vector.

Integrating positional encoding into embedding. The in-depth view of positional encoding shows the data flow: word embeddings, the positional encoding process, and the final embeddings. We show the process from the perspective of the overall matrix and the local cells (G1, G2). When users hover over an element value, the system directs attention to the operation of the cell, as depicted in Fig. 4.

5.1.4 Encoders and decoders

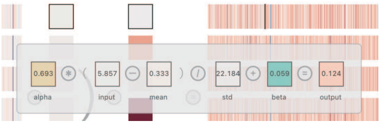
After the text is transformed into the final embedding matrix, it enters the architecture consisting of six encoder or decoder blocks. Six blocks are presented in a top-down flowchart. Each encoder or decoder block shares an identical internal structure. The flowchart illustrating the detailed structure is displayed on the side (Fig. 1C3). The attention operation is presented with focused clarity. The principle of residual concatenation is easy to understand, so only a combination of text and operators is used to show it. The presentation of layer normalization abstracts the data flow. The feed-forward network is described textually, highlighting its main components. The flowchart also serves as an interface for the user to interactively explore the details of the data flow and implementation process of each module (G2). Users can click and slide the view area to access the first-level unfolding views.

Explaining self-attention mechanism. This view outlines the process of data transformation within the multi-head attention mechanism. As shown in Fig. 1D, we utilize heatmap matrices to visualize the data distribution of the input, multi-head attention matrix, and output. The matrix dimensions and corresponding information are represented by the text. Additional views are employed to elucidate the intermediary processes of the attention mechanism. As illustrated in Fig. 5, users can access the corresponding explainable view by clicking on the prompted module (G2).

- **Emphasizing the attention head projection.** This view (Fig. 5B) emphasizes the process of generating Q,K and V through linear algebraic operations. Q,K and V matrices are learned through the training process and consist of weight matrices and noise, visually represented in the yellow-green color scheme. The input and output data are represented using red and blue color scales, respectively. Upon clicking on any element in the Q,K and V matrices, the system highlights the other related matrix elements, providing users with a clear and intuitive understanding of the calculation process for high-dimensional data.

- **Breaking down the attention operation.** To provide a clear and structured description of the implementation process, we explain the phases involved, namely matrix multiplication, scaling and masking, and softmax operation (Fig. 5C). Each operation supports an animation in which a single unit moves with the calculation steps. Automated processes help users understand the computing process and reduce the cost of user understanding. Additionally, users can click on the specific module for detailed explanations. Within each stage, users can delve into the numerical transformation process by clicking on an element in each matrix.

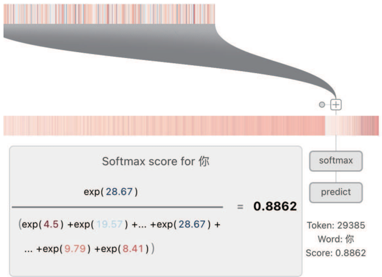
- **Representing the concatenation and linear projection.** Fig. 5D displays the process of subspace concatenation and linear projection in a linear algebraic manner. To enhance the intuitive understanding of the subspace stitching method, users can hover the mouse over a specific part of the concatenation result to reveal the corresponding subspace matrix.



Detailing the layer normalization. This part illustrates how layer normalization alters the numerical distribution of input data. The view is presented in an overview (Fig. 1E) and detailed way using matrix operations (G1,G2). Users can obtain detailed information on each element by hovering.

Decomposing the feed-forward network. The composition and computation of the feed-forward network are illustrated clearly in Fig. 1F. Independent single embedding vectors enter two linear transformations to obtain the output (G2). Each embedding vector operates independently and in parallel, affording users the flexibility to switch between different vectors by hovering over them. To provide a more detailed understanding of the activation function, users can dynamically demonstrate the ReLU operations by moving the mouse over the embedding vector and collaborating with cells.

5.1.5 Output: Generate output probabilities



Result preprocessing. By clicking on the "Linear & Softmax" module in Fig. 1C4, users can access the computation for the current decoder time step. This view includes a linear projection, a softmax function, and the predicted output. The linear projection is similar to the visual representation described ear-

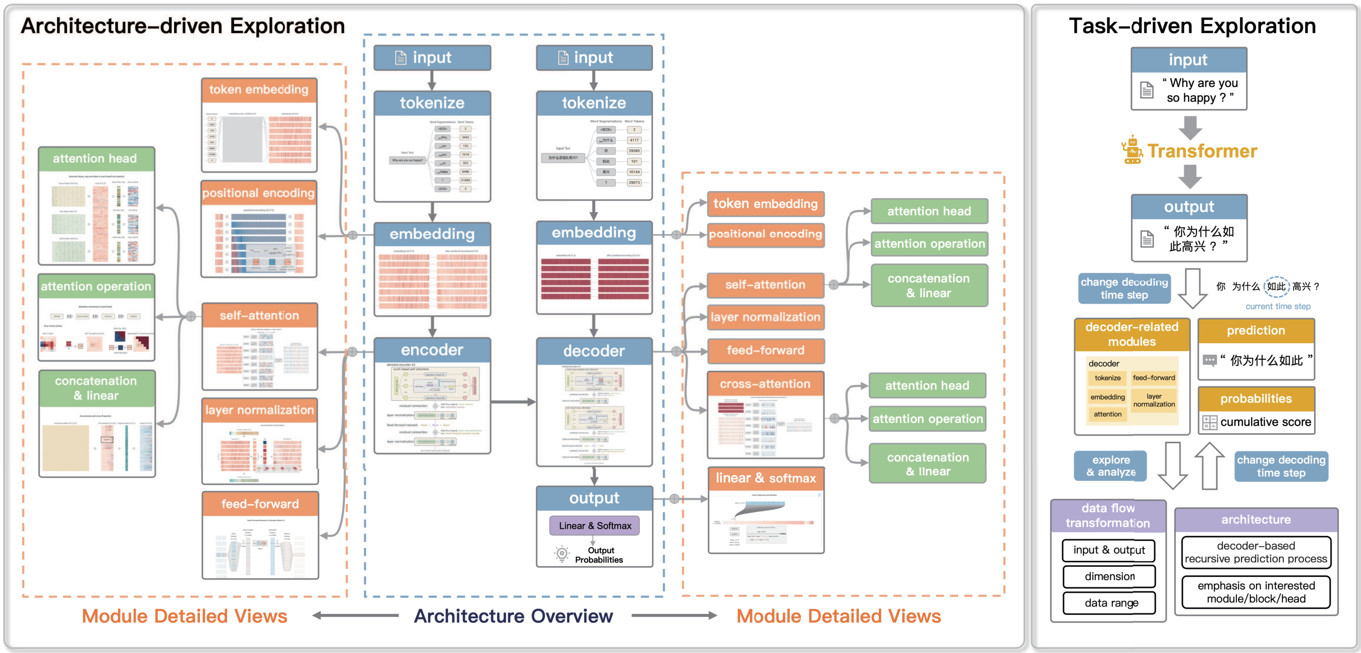


Fig. 3: The workflow of TransforLearn. We introduce the system from two aspects: **architecture-driven exploration** and **task-driven exploration**. In the **architecture-driven exploration**, the system provides an overview of the Transformer architecture and presents the detailed modules. There is a hierarchical relationship between the overview and the detailed modules: blue tabbed views are the topmost structures of the Transformer; orange tabbed views are the first-level unfolding state, and green tabbed views show the second-level detailed operations. Users can drill down from architecture overview to module detailed views by specific interactions. In the **task-driven exploration**, users will have a deeper understanding of the data flow transformation and model structure with the help of actual downstream tasks (machine translation in this system). By changing the decoding time step, users can discover the changes in data flow and final output results within the module.

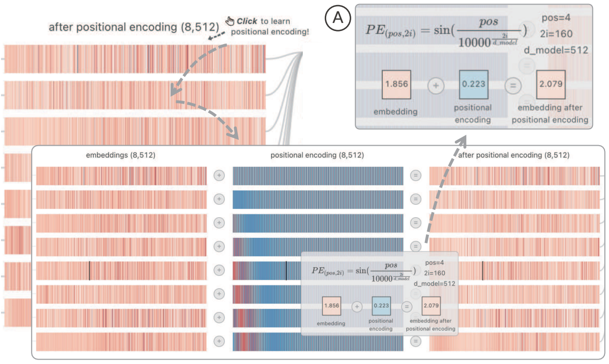

after positional encoding (8,512)
Click to learn positional encoding!
A
$PE_{(pos,2i)} = \sin(\frac{pos}{10000^{\frac{2i}{d\_model}}})$
pos=4
2i=160
d_model=512
1.856
0.223
2.079
embedding
positional encoding
embedding after positional encoding
embeddings (8,512)
positional encoding (8,512)
after positional encoding (8,512)

Fig. 4: A drill-down view of the positional encoding process for embedding. The focused tool tip displays the formula and parameter values for calculation upon selecting an element.

lier. The line is a metaphor for how linear transformations are projected. Users can click on the softmax button to view the calculation formula and the resulting probability of the current token. The system uses text to interpret information about the predicted words of the current decoding time step.

Prediction and Probabilities. The translation view presents four pieces of information: the complete translation output, the translated result and predicted result obtained at the current decoding time step and the cumulative probability. Users modify the decoding time step in the dialog box to view the transformation of decoder input, translated result, and cumulative probability.

5.1.6 Design alternatives

In this section, we delve into the visualization of one-dimensional vectors and two-dimensional matrices within the system, while also exploring the merits and drawbacks associated with the utilization of color pixel blocks alongside other design alternatives. As a visual tutorial, the system design should restore the underlying features of the model to the greatest extent and be easy for beginners to understand.

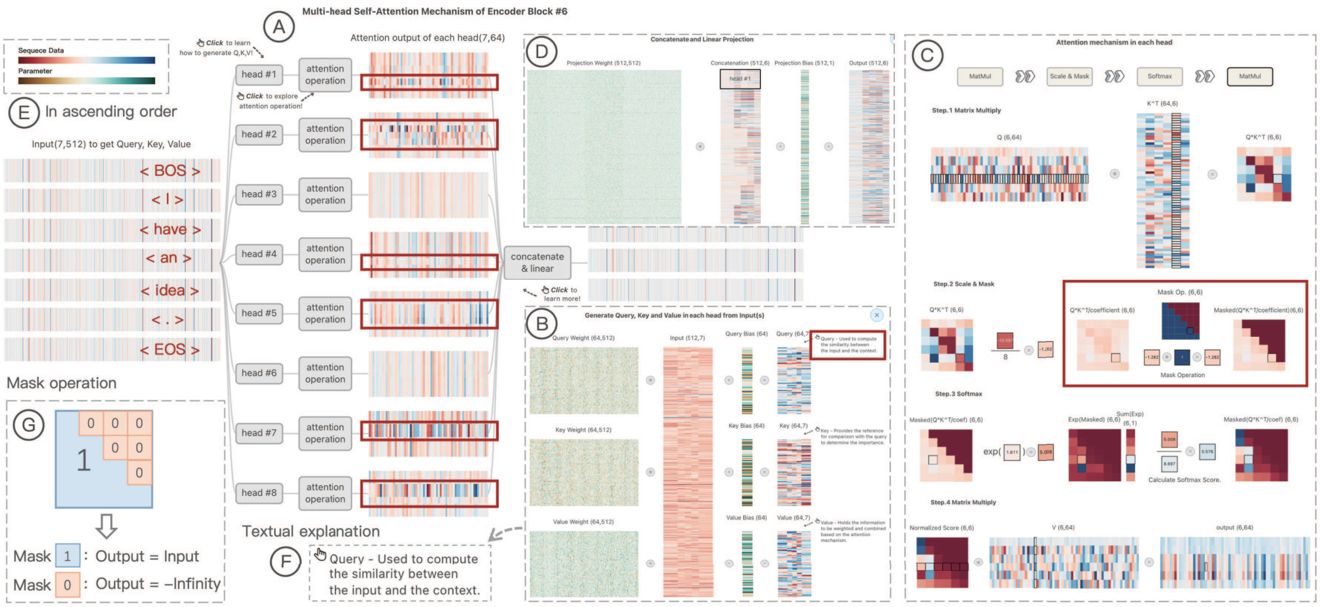
For one-dimensional vector visualization, commonly used techniques include height, polyline, and color mapping. Despite its intuitiveness, height mapping has low information density. Polylines illustrate element trends well but can add visual clutter and decrease information density. Colors, however, can represent a large number of vector elements in a confined space, vividly displaying data dimensions. Given embedding's complexity, we chose color mapping for efficient information conveyance.

The attention mechanism, involving two-dimensional matrix computation, can be visualized using color-coded pixel lattice matrices. Each matrix element is represented as an independent pixel, conveying the matrix's actual layout and meaning. An alternative approach is a chordal graph, like Dodrio [51], where attention's semantics are presented through nodes and relationship edges. But this design increases visual complexity with a large number of elements, creating structural layout challenges.

5.2 Task-driven Exploration

In this section, we will introduce the interaction process of the system in the task-driven exploration (G3), as illustrated in Fig. 3. In the translation task, the model processes input English text and output the corresponding Chinese text. The process of handling the output result in the decoder end is non-parallel, generating individual words one by one to ensure the correct order of translation. The interactive design of this section fully considers the importance of non-parallelism in the decoder end. We interpret "translation progress" as the "decoding time step," and non-parallelized processing can be achieved by modifying the decoding time step. Specifically, when we input the sentence "Why are you so happy?", the Transformer will give the answer "你为什么如此高兴？ ". Users can change the decoding time step from "为什么" to "如此", and discover the changes and translation results related to the decoder.

Explore data flow changes. Data flow changes mainly involve input and output, data dimension, and data range (Fig. 6A). When users

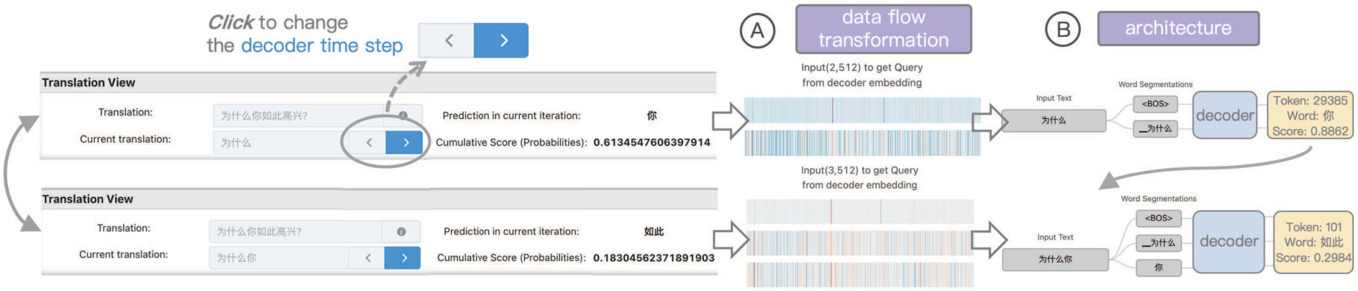


Query - Used to compute the similarity between the input and the context.

Key - Provides the reference for comparison with the query to determine the importance.

Value - Holds the information to be weighted and combined based on the attention mechanism.

Fig. 5: Explanation of the data transformation process in the multi-head attention mechanism **(A)** handling the input "I have an idea". The part selected by the red square is the processing of "have an idea" emphasized by multiple attention heads. The attention head view **(B)** shows how to generate Q, K and V in each head. Textual explanation **(F)** focuses on the text description selected by the box. The attention operation view **(C)** breaks down the process of computing the attention matrix. Users can turn to Mask operation **(G)** to explore its principle. The concatenation and linear view **(D)** show how the multi-head attention results are combined to generate the final result.



Translation View

Translation: 为什么你如此高兴?

Current translation: 为什么

Prediction in current iteration: 你

Cumulative Score (Probabilities): 0.6134547606397914

Translation View

Translation: 为什么你如此高兴?

Current translation: 为什么你

Prediction in current iteration: 如此

Cumulative Score (Probabilities): 0.18304562371891903

Fig. 6: Task-driven exploration: users can click on the icon in the Translation view (Fig. 1B) to change the decoder time step. **(A)** shows that the decoder time step changes the data dimension and range. **(B)** shows the recursive prediction process based on the decoder.

increase or decrease the current decoding time step, the system will either add or remove the input word and the corresponding output result. Users can perceive the differences in the color distribution and infer changes in the data range.

Analyze structural features. From Fig. 6B, we can see that the predicted word of this iteration enters the next time step as input. The prediction and cumulative possibilities in Fig. 1B show the current translation progress and predicted possibility. Meanwhile, when users explore the downstream tasks, we find that the iterative input of the decoder end, multiple block operations, and diverse attention heads may cause confusion for beginners in terms of concepts. Therefore, controlling the current decoding time step can help users focus on a specific module, block, or head that they are interested in.

5.3 Interaction between two exploration modes

The combination of architecture-driven exploration and task-driven exploration modes serves as a pathway from abstract understanding to practical implementation. Users are encouraged to freely switch between the two proposed modes. In cases where beginners find abstract visual metaphors confusing, TransforLearn prompts them to switch to the task-driven exploration mode and begin with real-world data. The task-driven exploration mode provides insights into data flow transformations and structural relationships from the perspective of specific task processing, thereby motivating beginners to return to the architecture-driven exploration mode. This iterative switching between modes aims to foster a comprehensive understanding of Transformer, ensuring that beginners can navigate both the abstract and practical aspects of the model effectively.

6 Usage Scenario

We illustrate usage scenarios informed by the interactions we had with the lecturers and students as well as their exploration of TransforLearn. We invited a student eager to learn about Transformers and observed his interactions. Here, we provide a scenario of “Self-study guidance for a beginner” (Fig. 5) based on his experience with the system, and put an additional hypothetical scenario of “Teaching aid for lectures” in the supplementary materials due to the page limit.

Rex, a senior undergraduate student, is eager to learn and leverage the Transformer model to sequence data predictions in sports. Despite grasping the model structure basics from blogs and videos, he still struggles with detailed structure understanding and translating theoretical knowledge into practical tasks. Therefore, he has taken an interest in TransforLearn, hoping to get guidelines from it.

Rex first follows the **task-driven exploration** mode and inputs the sentence "I have an idea". The resulting visualization of sequence vectors and textual descriptions (Fig. 1C2) allows him to understand the transformation of participles into long data sequences. This stimulates him to think about how to adapt his high-dimensional time-series data into Transformer.

Rex's primary purpose in utilizing Transformer is to extract features from sequence data. He has prior knowledge that the attention mechanism was crucial for feature extraction, but he remains uncertain about what the model precisely learns under this abstract concept. Following the **architecture-driven exploration** mode, he clicks on the multi-head self-attention module in the last encoder block (Fig. 5A), which provided visual insights into the data flow transformation and multi-head structure. However, he still has doubts regarding the concept and generation process of the Q, K, and V matrices. To gain further clarity, Rex clicks on a specific head button based on the text prompt. In Fig. 5B, combined with the color mapping in Fig. 5E, he discovers that the generation of matrices relied on trained weights and biases in the yellow to green color scheme. He says, *"The text description next to the matrix in Fig. 5F further helps me grasp the matrices concepts."* Additionally, when observing the attention score matrices, Rex notices that most heads exhibit stronger color patterns on "have an idea" (Fig. 5A). In particular, the 2nd, 5th and 8th heads of the last encoder block all have significant color changes. He comments that *"Different head has its own focus when processing information."* These findings lead Rex to switch back to the **architecture-driven exploration** mode, where he could further explore the role of attention mechanisms. Rex next ponders the use of decoders for prediction. Although initially confused by the interference of post-order information, he understands their handling thanks to the lower triangular matrix in the mask operation (Fig. 5G).

7 Evaluation

We identified four key features that are crucial for the design of a visual tutorial tool for Transformers, as depicted in Tab. 1. In Sec. 3.2, we list several novel visualization systems involving Transformers. As Tab. 1 demonstrates, these systems often lack an explanation of the overall model architecture and the underlying mathematics, as they tend to focus on specific aspects of the models.

We categorized existing tutorials into three primary types: blogs, videos, and code explanations, and examined illustrative instances from each category (Tab. 1). Due to the limitations of their medium, i.e., one-way communication leading to a passive learning style, these three forms do not offer immersive learning experiences. Furthermore, how to combine abstract concepts with concrete tasks, such as machine translation, is a source of confusion for many learners.

Table 1: Supporting key features (G1, G2, G3 and G4) of tutorial tools Performances of tutorial tools on key features. Blogs are primarily feature blogs by Jay Alammar [2,3]. The representative work of the video is *Transformer Neural Networks* [9]. The work of code interpretation is *The Annotated Transformer* [33]. Other Vis System mainly includes Attviz [62], Attention flows [11], and Dodrio [51].

Feature Work	Model Overview	Detailed Computation	Task Alignment	Engaging Interface
Blog	✓	✓	✗	✗
Video	✓	✓	✗	✗
Code	✗	✗	✓	✗
Other Vis System	✗	✗	✓	✓
TransforLearn	✓	✓	✓	✓

To evaluate the effectiveness and usability of our work, we further conducted an in-person study and interviews with potential learners.

7.1 Experiment Setup

7.1.1 Participants

TransforLearn targets users possessing a foundational understanding of deep learning and NLP tasks, aiming to master Transformers. We enlisted 18 student participants (9 female, 9 male; 6 undergraduates, 12 postgraduates) from a university's computer science department. A comprehensive questionnaire was administered to assess their knowledge of related concepts using a five-point Likert scale. Results showed that most of them had acquired machine learning and visualization fundamentals. All displayed an eagerness to learn about Transformer with TransforLearn.

Table 2: Objective questions. The first column of the table is the difficulty levels, which are classified as easy, medium and hard. The second column is the targets of questions corresponding to our design goals. The third column is a brief description of the questions. FFN refers to Feed-forward Network, Add & LN refers to Residual connection and layer normalization, and PE refers to Position encoding.

Level	Goal	Question
easy	G1	Q1: Components and data flow of feed-forward network.
easy	G3	Q2: Identify key words from attention matrix.
easy	G3	Q3: Final output in translation task and its derivation.
medium	G1	Q4: Differences between cross- and self-attention.
medium	G2	Q5: Add & LN significance and implementation.
medium	G1	Q6: Parallelism in Transformer.
hard	G2	Q7: Reasons for scaling before softmax.
hard	G2	Q8: Process of calculating PE & variation with position.

16 participants with no prior Transformer experience and an average score below 3 are considered beginners in deep learning due to their limited knowledge in this field. We further divided them randomly into two groups of 8 participants each, B1-8 (Group B) and T1-8 (Group T), to learn Transformer through the "blog" and "TransforLearn" respectively. As discussed in Sec. 3.3, the blogs by Jay Alammar [3] and the one provided by Harvard [33] have earned a reputation for their authority and popularity. This is reflected in their substantial video views—Alammar's blog has even accrued up to 160,000 views—and their consistent ranking within the top 10 on search engines. The remaining two participants' scores with an average score higher than 3 were referred to as E1-2 (Group E), where E stands for "expert". Despite not being the primary users of TransforLearn, we allowed them to partake in exploration similar to Group T, to gain diverse insights into TransforLearn's strengths and weaknesses.

7.1.2 Procedure

We conducted individual, in-person studies with participants in an offline environment. Participants used their personal computers and browsers to master the Transformer via blog (Group B) or the TransforLearn (Group T and Group E) and subsequently provided feedback. We mainly compare Group B and Group T, with Group E following the same procedure as Group T. With informed consent, we captured audio and screen recordings for later analysis.

The study consisted of three sessions, starting with a 5-minute introduction wherein participants were acquainted with the fundamentals of the Transformer. For Group T, we additionally provided a 5-minute tutorial outlining the various views and workflow of TransforLearn instead of giving participants time to explore. Subsequently, all participants answered a series of objective questions upon concluding their exploration, encouraged to "think aloud" and ask questions. To assess beginners' learning efficiency, we unobtrusively timed their responses, encouraging thoughtful problem-solving without creating any time pressure. After the objective test, Group B finished their session, while Group T further explored TransforLearn, answered an exit questionnaire, and participated in a final interview for detailed feedback. The study duration was around 30 minutes for Group B and 50 minutes for Group T, with each participant receiving a $10 reward.

7.1.3 Test questions and measurement

During the evaluation process, users adopt TransforLearn to solve the machine translation task. In addition to employing two questionnaires that utilized a five-point Likert scale—one at the study's onset to profile users, and another at its conclusion to assess subjective feelings—we also devised a series of objective questions varying in difficulty from our design goals Tab. 2[1]. Given the complexity and

[1] Please refer to Supplementary Materials for all subjective or objective questions, participants' responses, and evaluation criteria for objective questions.

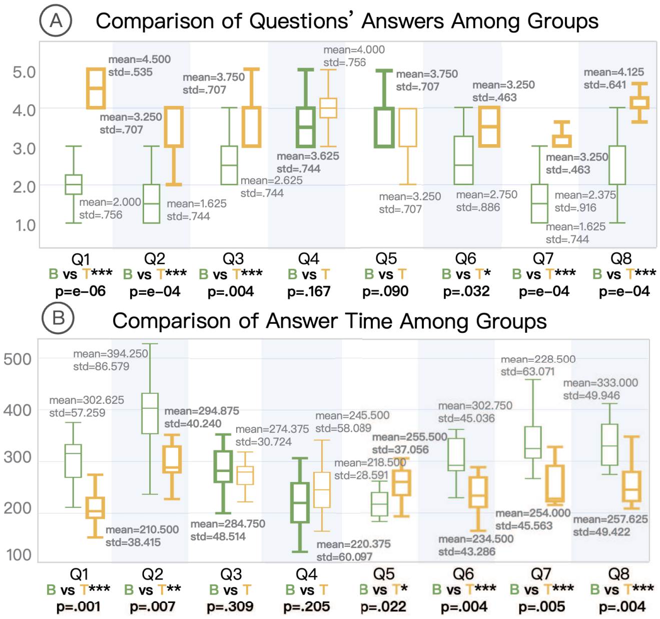
inherent probabilistic uncertainty of neural networks, it is challenging to pose questions with unique, definitive answers akin to those used in past algorithm visualization evaluations [19]. We gathered common Transformer-related recruitment interview questions from technology companies via the internet and selected eight questions of varying difficulty levels, corresponding to **G1-G3**, to evaluate whether the predetermined design goals of TransforLearn had been achieved. We assigned scores of 1-5 based on the scoring system and documented the users' answers. The scoring system follows the Structure of the Observed Learning Outcome (SOLO) taxonomy [5], a model that describes the increasing complexity of learners' understanding. The scoring system mainly adopts the first three levels in SOLO, namely pre-structural, uni-structural and multi-structural, and maps them linearly to the Likert scale. The question sequence is designed to progressively increase in difficulty, enabling users to enhance their familiarity with the system as they proceed. Statistical results analysis by unpaired T-test was performed to facilitate objective measurement. Furthermore, we conducted a comprehensive analysis of the detailed answer texts to extract additional information.

7.2 Results and Analysis

7.2.1 Objective experiment results

We performed unpaired t-tests to compare the significance of the difference in the accuracy and the completion time of objective questions between the two groups. As Fig. 7A illustrated, TransforLearn improves mean accuracy in most cases except for Q5 (3.750/3.250 for Group B/T), with different significance levels ($p < 0.005$ for Q1-3 and Q7-8, $p < 0.05$ for Q6). As for completion time (Fig. 7B), Group T spent less time significantly in Q1/6/7/8 ($p < 0.005$) and Q2 ($p < 0.01$). Additionally, we calculated the learning efficiency index [39] using the following formula $E_{GroupX,i} = \frac{Score_{GroupX,i}}{Time_{GroupX,i}} * 100$. Fig. 7C presents $E_{GroupX,i}$ of all participants, highlighting the superior performance of Group T over Group B. Analyzing both the statistical results and the process of finding answers between the two groups, we verify the effectiveness of TransforLearn and report here key findings.

TransforLearn generally improves users' understanding of architecture and tasks more than blogs. TransforLearn encompasses the entire architecture from a breadth perspective, whereas blogs omit feed-forward networks. Consequently, in Q1, Group T significantly outperformed Group B ($p = 2.24e-6$). Participants T1, T3-4, and T6 accurately identified the feed-forward neural network composition as "Linear + ReLU + Linear," with a data dimension of "512-2048-512." Conversely, Group B provided vague responses such as "Input layer + Hidden layer + Output layer" and "dimension first increases, then decreases." In terms of depth, the more challenging questions (Q7 and Q8) yielded less clear answers. Group T managed to deduce the question's intent based on the system interface's information changes and attempted a response, whereas Group B was completely confused due to the blog's lack of explanation. Moreover, when confronted with Q2, Group T observed attention matrices and pinpointed significant words via heat-maps, highlighting our system's ability to better correlate the model with the task. Group B, however, almost failed due to the limitation of presentation in blogs. From the results of Q1-2 and Q7-8, we confirmed that TransforLearn meets the design goals and thus improves user understanding.



Ⓒ Comparison of Learning Efficiency Index Among Groups

$E_{GroupX,i}$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$	Mean	Std
$X = B$	0.737	1.005	0.680	0.839	0.981	0.824	0.965	0.851	0.851	0.121
$X = T$	1.421	1.702	1.631	1.402	1.263	1.385	1.381	1.542	**1.466**	0.146

Fig. 7: Data insights of objective experiment results. The boxplots depicted in green are for Group B and those in yellow represent Group T's results. The x-axis signifies the eight experiment questions. The y-axis in **(A)** signifies the scores ranging from 1 to 5 (higher is better). The y-axis in **(B)** signifies the answer time ranging from 100 to 500 (seconds). Mean, standard deviation, and T-test p-value (indicated by *, **, *** for $p < .05, .01$, and .005, respectively) are also presented. (C) compares the Learning Efficiency Index between two Groups.

TransforLearn brings more activity, autonomy, and divergent thinking through an interactive interface. Group T, who engaged with TransforLearn, thought aloud and actively posed questions while problem-solving. In addressing Q7, T3, despite lacking a comprehensive mathematical understanding, remarked, *"Although I'm unsure about the attention matrix's need for $\sqrt{d_k}$ scaling, I observed smaller attention values post-scaling. Given my basic understanding of Softmax, it may prevent gradient disappearance."* Although Group B scored slightly higher on Q5, it doesn't necessarily suggest blog superiority. Their success on Q5, where the blog gave a clear answer about "meaning", stemmed from reliance on the blog content rather than independent thought, limited by the blogger's writing style. Conversely, Group T described the implementation based on their observations, although their answers concerning "meaning" were skewed or incomplete as they were solely based on personal understanding. This is also the reason why Group T spent significantly more time on this question than Group B ($p = 0.022$). Q5 and Q7 responses underscore the advantages of an interactive interface in enhancing user activity.

A broader coverage and enhanced interaction support efficiency in learning. The T-test results indicate that Group T surpasses Group B not only in accuracy but also in time spent for Q1-2 and Q7-8. Among them, Group B's process of answering Q1 warrants further discussion. The blog mentioned feed-forward networks but lacked a detailed introduction, leading Group B to believe they had learned the concept falsely. They spent considerable time rechecking the blog, only to realize they had to guess the answer based on prior knowledge. We suspect that the blog's dense information hindered users from identifying key points, which was not an issue with visual and interactive TransforLearn. Additionally, a significant difference in elapsed time emerged between the two groups for Q6 ($p = 0.004$), despite no significant difference in accuracy. Follow-up interviews revealed that Group B focused on the attention mechanism highlighted in the blog but struggled to understand the model holistically. In contrast, Group T quickly answered Q6 after gaining a comprehensive understanding after architecture-driven exploration. Comparing the time spent on Q1-2 and Q6-8 shows that TransforLearn doesn't overwhelm beginners with complex systems and extensive exploration. Instead, the dual exploration modes help users gain knowledge more efficiently and confidently.

7.2.2 Subjective experiment results

Based on the analysis of the exit questionnaire and the interviews, we evaluated participants' subjective feedback on TransforLearn and de-

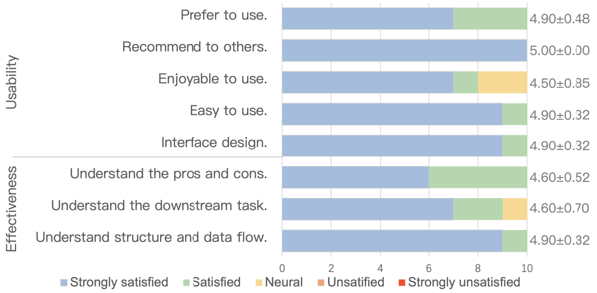


Fig. 8: Results from the subjective questionnaires. The stack bars indicate feedback scores and the rightmost column shows $Mean \pm STD$.

rived valuable insights. Notably, two participants from Group T hold the view that the interface provides substantial guidance, while the flow chart and hierarchical structure are deemed clear and logical. For details, please refer to Supplementary Materials.

Usability and effectiveness. As depicted in Fig. 8, TransforLearn received high ratings in both usability and effectiveness. Participants expressed willingness to continue exploring the Transformer with TransforLearn and recommended it to other beginners. Half of the participants noted that TransforLearn's ability to provide operation details absent from blog posts was particularly beneficial. Furthermore, E1 and T2-4 mentioned that the combination of architecture-driven and task-driven exploration methods facilitated deeper understanding.

Validating the knowledge for experts. Group E, as expected, performed exceptionally on objective questions, displaying enriched thinking and knowledge verification. When E1 addressed Q2, he noticed the relationship between multi-head and single-head attention in dimensions. He entered the sentence *"Why are you so happy?"* and questioned whether the "8" in the single-head attention dimension (8, 64) referred to sequence length or the number of heads. Through task-driven exploration and translation iteration adjustments, E1 confirmed it represented sequence length. He commented that *"It used to be my initial misconceptions when I started to learn Transformer. TransforLearn's rich interaction easily helps beginners verify that the single-head space resembles a subspace learning process."* When E2 tackled Q8, she considered using TransforLearn to verify the embedding results of two identical words in a sentence to enhance the understanding of positional encoding. E2 input *"Why do you think you are right?"* and observed that the two instances of "you" initially shared the same embeddings, which became distinct following positional encoding.

Different appropriate learning resources for different needs. Although group T performed well in our learning tasks, TransforLearn may not cater to all learning needs, as remarked by E2. 1) Casual learners seeking a general overview of Transformer can utilize TransforLearn, but may quickly browse blogs and videos for convenience. 2) Those aiming for detailed knowledge and application of Transformers will find our tool aligned with their needs, as outlined in Fig. 2B. 3) For advanced learners desiring a deep theoretical understanding or further interpretation, TransforLearn aids visual validation, but deeper study will require academic papers or specific XAI tools.

Need for more instructions, animations, and comparisons. The majority of the suggestions received emphasized the desire for more diverse and comprehensive visualizations. T3, who correctly guessed the answer to Q7, suggested that *"If it provides a visualization of the follow-up results without scaling the attention and compares it with results with scaling, I would have been more confident in my guess."* Moving forward, we will consider implementing more comparative visualizations of crucial mechanisms, such as comparisons between attention heads, layers, and pre- and post-finetuning, to facilitate rapid knowledge testing and verification for users.

8 Discussion

Alleviating visual interference from color mapping. To minimize color fluctuations resulting from information overload in pixel bar charts, it's crucial to select an appropriate color palette. The chosen color scheme should span various domains to effectively highlight data features. Additionally, alternatives like using shapes or textures can enhance the visualization of underlying patterns in the data [61]. The pixel bar chart employs straightforward color usage to maintain a low cognitive load. Moreover, interactive elements are integrated, enabling users to customize the visualization as needed. In forthcoming research, our focus is on investigating visualization techniques that not only present actual data with minimal cognitive load but also reduce visual distractions.

Trade-off between the presentation of operation details and semantic understanding of decision processes. TransforLearn has limited capabilities in connecting model parameters with textual information and semantically understanding model decisions process, influenced by target audience and task objectives, as well as model architecture and data types. Visual analytics systems targeting deep learning experts typically do not display detailed architectures, operations, and parameters. Instead, they focus on enhancing information density by transforming parameters like embeddings and attentions into semantically understandable data, such as word importance and semantic score [54, 55, 60]. Moreover, due to the high perceptibility of images and relatively simple model architecture, works such as CNN Explainer [52] effectively demonstrate model details while conveying the semantic process of handling images by presenting an activation map. To help non-experts comprehend Transformers, we inevitably sacrifice some ability to tightly integrate input text with model decisions, for instance, by displaying raw embeddings rather than summarizing and connecting them to words via interpretability methods. We plan to explore richer visualization forms to address this issue.

Generalizing to other Transformer variants, tasks, and modalities. TransforLearn has proven effective in the learning process. We can easily expand the *Architecture Overview* to include different architecture-level variants, such as Encoder-only models (e.g., BERT) and Decoder-only models (e.g., GPT). Similarly, we can modify the *Module Detailed View* to accommodate other module-level variants [26], such as changes to attention mechanisms, multi-head mechanisms, and positional encoding, among others. This generalization process also supports various tasks within the NLP domain. For Transformer variants employed in other modalities, supplementary components may be necessary depending on the input data type, including the addition of visualizations for images, speech, and audio.

In-depth and longitudinal evaluation of benefits. Our user study, which combines qualitative and quantitative analysis, serves as an initial step in showcasing the educational benefits of TransforLearn. To further assess its usability and implement subsequent improvements, we need to broaden the evaluation's scope and duration, encompassing a wider array of users. Drawing from Gan Lab's valuable experience in long-term evaluation [21, 22], we plan to deploy TransforLearn on the web shortly and gather interaction logs from various user types [24] This will enable us to conduct a more in-depth analysis of their understanding and engagement levels.

9 Conclusion

In conclusion, we present TransforLearn, an innovative interactive visual tutorial tool for deep learning beginners and non-experts to understand the complex structure and abstract data representation of the Transformer model. TransforLearn adopts two exploration modes to provide multi-level structure displays and help users understand the working process of the downstream task. Our expert panel and meticulously devised controlled user study ascertain that TransforLearn is conducive to effective, immersive, and self-guided learning.

10 Acknowledgments

The authors want to thank reviewers for their suggestions. This work is supported by the Natural Science Foundation of China (NSFC No.62202105) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0103), General Program (No. 21ZR1403300), Sailing Program (No. 21YF1402900) and ZJLab.

References

[1] S. Abnar. From attention in transformers to dynamic routing in capsule nets. https://samiraabnar.github.io/articles/2019-03/capsule, 2019. Last accessed on July 1, 2023.

[2] J. Alammar. A visual guide to using bert for the first time. https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/, 2019. Last accessed on July 1, 2023.

[3] J. Alammar. The illustrated retrieval transformer. https://jalammar.github.io/illustrated-retrieval-transformer/, 2022. Last accessed on July 1, 2023.

[4] G. Alicioglu and B. Sun. A survey of visual analytics for explainable artificial intelligence methods. *Computers & Graphics*, 102:502–520, 2022.

[5] J. B. Biggs and K. F. Collis. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press, 2014.

[6] P. Bloem. Transformers from scratch. https://peterbloem.nl/blog/transformers, 2019. Last accessed on July 1, 2023.

[7] A. M. P. Braşoveanu and R. Andonie. Visualizing transformers for nlp: A brief survey. In *2020 24th International Conference Information Visualisation (IV)*, pp. 270–279, 2020.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[9] CodeEmporium. Transformer neural networks - explained! (attention is all you need). https://www.youtube.com/watch?v=TQQlZhbC5ps, 2022. Last accessed on July 1, 2023.

[10] A. Coenen, E. Reif, A. Yuan, B. Kim, A. Pearce, F. Viégas, and M. Wattenberg. Visualizing and measuring the geometry of bert, 2019. doi: 10.48550/ARXIV.1906.02715

[11] J. F. DeRose, J. Wang, and M. Berger. Attention flows: Analyzing and comparing attention mechanisms in language models, 2020. doi: 10.48550/ARXIV.2009.07053

[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[14] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. In *Computer Graphics Forum*, vol. 36, pp. 458–486. Wiley Online Library, 2017.

[15] C. Guan, X. Wang, Q. Zhang, R. Chen, D. He, and X. Xie. Towards a deep and unified understanding of deep neural models in NLP. In K. Chaudhuri and R. Salakhutdinov, eds., *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 2454–2463. PMLR, 09–15 Jun 2019.

[16] D. Gunning. Explainable artificial intelligence (xai) darpa-baa-16-53. *Defense Advanced Research Projects Agency*, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[18] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2674–2693, 2019. doi: 10.1109/TVCG.2018.2843369

[19] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3):259–290, 2002.

[20] Z. Jin, X. Wang, F. Cheng, C. Sun, Q. Liu, and H. Qu. ShortcutLens: A visual analytics approach for exploring shortcuts in natural language understanding dataset. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2023. doi: 10.1109/tvcg.2023.3236380

[21] M. Kahng and D. H. Chau. How does visualization help people learn deep learning? evaluation of gan lab. In *IEEE VIS 2019 Workshop on EValuation of Interactive VisuAl Machine Learning Systems*, 2019.

[22] M. Kahng and D. H. P. Chau. How does visualization help people learn deep learning? evaluating gan lab with observational study and log analysis. In *2020 IEEE Visualization Conference (VIS)*, pp. 266–270. IEEE, 2020. doi: 10.1109/TVCG.2018.2864500

[23] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viegas, and M. Wattenberg. GAN lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):310–320, jan 2019. doi: 10.1109/tvcg.2018.2864500

[24] Y. Li, Y. Qi, Y. Shi, Q. Chen, N. Cao, and S. Chen. Diverse interaction recommendation for public users exploring multi-view visualization using deep learning. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):95–105, 2023. doi: 10.1109/TVCG.2022.3209461

[25] Z. Li, X. Wang, W. Yang, J. Wu, Z. Zhang, Z. Liu, M. Sun, H. Zhang, and S. Liu. A unified understanding of deep nlp models for text classification, 2022. doi: 10.48550/ARXIV.2206.09355

[26] T. Lin, Y. Wang, X. Liu, and X. Qiu. A survey of transformers. *AI Open*, 2022. doi: 10.1016/j.aiopen.2022.10.001

[27] D. Liu, W. Cui, K. Jin, Y. Guo, and H. Qu. Deeptracker: Visualizing the training process of convolutional neural networks, 2018. doi: 10.48550/ARXIV.1808.08531

[28] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):91–100, 2016. doi: 10.1109/TVCG.2016.2598831

[29] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective, 2017. doi: 10.48550/ARXIV.1702.01226

[30] D. Long and B. Magerko. What is ai literacy? competencies and design considerations. In *2020 ACM CHI Conference on Human Factors in Computing Systems, CHI 2020*, p. 3376727. Association for Computing Machinery, 2020. doi: 10.1145/3313831.3376727

[31] T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3428–3448. Association for Computational Linguistics, Florence, Italy, Jul 2019. doi: 10.18653/v1/P19-1334

[32] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, feb 2018. doi: 10.1016/j.dsp.2017.10.011

[33] H. NLP. The annotated transformer. https://nlp.seas.harvard.edu/2018/04/03/attention.html, 2018. Last accessed on July 1, 2023.

[34] C. Olah. Olah's blogs. http://colah.github.io/, 2020. Last accessed on July 1, 2023.

[35] K. O'Shea and R. Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[36] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022. doi: 10.48550/ARXIV.2203.02155

[37] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[38] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[39] G. S. Bruce. Chapter 15 - learning efficiency goes to college. In D. J. Moran and R. W. Malott, eds., *Evidence-Based Educational Methods*, Educational Psychology, pp. 267–275. Academic Press, San Diego, 2004. doi: 10.1016/B978-012506041-7/50016-4

[40] C. Seifert, A. Aamir, A. Balagopalan, D. Jain, A. Sharma, S. Grottel, and S. Gumhold. Visualizations of deep neural networks in computer vision: A survey. *Transparent data mining for big and small data*, pp. 123–144, 2017. doi: 10.1007/978-3-319-54024-5_6

[41] Z. Shao, S. Sun, Y. Zhao, S. Wang, Z. Wei, T. Gui, C. Turkay, and S. Chen. Visual explanation for open-domain question answering with bert. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–18, 2023. doi: 10.1109/TVCG.2023.3243676

[42] A. Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018.

[43] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg. Direct-manipulation visualization of deep networks, 2017. doi: 10.48550/

ARXIV.1708.03788

[44] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks, 2016. doi: 10.48550/ARXIV.1606.07461

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[46] J. Vig. Bertviz: A tool for visualizing multi-head self-attention in the bert model. *ICLR Workshop: Debugging Machine Learning Models*, 05 2019. doi: 10.18653/v1/. D17-2021.

[47] J. Vig. A multiscale visualization of attention in the transformer model. *CoRR*, abs/1906.05714, 2019. doi: 10.18653/v1/p19-3007

[48] J. Wang, L. Gou, H.-W. Shen, and H. Yang. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):288–298, 2019. doi: 10.1109/TVCG.2018.2864504

[49] J. Wang, Y. Li, Z. Zhou, C. Wang, Y. Hou, L. Zhang, X. Xue, M. Kamp, X. Zhang, and S. Chen. When, where and how does it fail? a spatial-temporal visual analytics approach for interpretable object detection in autonomous driving. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–16, 2022. doi: 10.1109/TVCG.2022.3201101

[50] X. Wang, J. He, Z. Jin, M. Yang, Y. Wang, and H. Qu. M2lens: Visualizing and explaining multimodal models for sentiment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):802–812, jan 2022. doi: 10.1109/tvcg.2021.3114794

[51] Z. J. Wang, R. Turko, and D. H. Chau. Dodrio: Exploring transformer models with interactive visualization. *CoRR*, abs/2103.14625, 2021.

[52] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, and D. H. P. Chau. CNN explainer: Learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1396–1406, feb 2021. doi: 10.1109/tvcg.2020.3030418

[53] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):1–12, 2018. doi: 10.1109/TVCG.2017.2744878

[54] J. Xia, L. Huang, W. Lin, X. Zhao, J. Wu, Y. Chen, Y. Zhao, and W. Chen. Interactive visual cluster analysis by contrastive dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):734–744, 2023. doi: 10.1109/TVCG.2022.3209423

[55] J. Xia, F. Ye, W. Chen, Y. Wang, W. Chen, Y. Ma, and A. K. Tung. Ldsscanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):236–245, 2018. doi: 10.1109/TVCG.2017.2744098

[56] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.

[57] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015.

[58] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018. doi: 10.1109/MCI.2018.2840738

[59] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning, 2020. doi: 10.48550/ARXIV.2008.09632

[60] Y. Zhao, L. Ge, H. Xie, G. Bai, Z. Zhang, Q. Wei, Y. Lin, Y. Liu, and F. Zhou. Astf: visual abstractions of time-varying patterns in radio signals. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):214–224, 2023. doi: 10.1109/TVCG.2022.3209469

[61] Y. Zhao, X. Wang, C. Guo, M. Lu, and S. Chen. Contextwing: Pair-wise visual comparison for evolving sequential patterns of contexts in social media data streams. *Proc. ACM Hum.-Comput. Interact.*, 7(CSCW1), apr 2023. doi: 10.1145/3579473

[62] B. Škrlj, N. Eržen, S. Sheehan, S. Luz, M. Robnik-Šikonja, and S. Pollak. Attviz: Online exploration of self-attention for transparent neural language modeling, 2020. doi: 10.48550/ARXIV.2005.05716

A Supplemental Materials

We provide the following supplementary materials:

- **Supplementary Material for TransforLearn Evaluation.** In this material, we submit 3 specific supporting documents: Preliminary Study for Requirement Analysis, Evaluation Supplementary Material for TransforLearn, and Usage Scenario Supplementary Material for TransforLearn. Among them, Preliminary Study for Requirement Analysis describes the process of learning challenge and needs analysis, mainly recording the content and results of the questionnaire. Evaluation Supplementary Material for TransforLearn provides a detailed description of the evaluation process and the questions we set. Additionally, we have included the results of our user research to facilitate readers' understanding. In Usage Scenario Supplementary Material for TransforLearn, we record interviews with teachers and supplement hypothetical classes as teaching aids.
- **Video Material for TransforLearn.** We provide a video introduction to our work on TransforLearn. In the video, readers can gain a better understanding of TransforLearn and our efforts.