How Does Attention Work in Vision Transformers? A Visual Analytics Attempt

Yiran Li, Junpeng Wang, Xin Dai, Liang Wang, Chin-Chia Michael Yeh, Yan Zheng, Wei Zhang, and Kwan-Liu Ma, *Fellow, IEEE*

***Abstract*—Vision transformer (ViT) expands the success of transformer models from sequential data to images. The model decomposes an image into many smaller patches and arranges them into a sequence. Multi-head self-attentions are then applied to the sequence to learn the attention between patches. Despite many successful interpretations of transformers on sequential data, little effort has been devoted to the interpretation of ViTs, and many questions remain unanswered. For example, among the numerous attention heads, which one is more important? How strong are individual patches attending to their spatial neighbors in different heads? What attention patterns have individual heads learned? In this work, we answer these questions through a visual analytics approach. Specifically, we first identify *what* heads are more important in ViTs by introducing multiple pruning-based metrics. Then, we profile the spatial distribution of attention strengths between patches inside individual heads, as well as the trend of attention strengths across attention layers. Third, using an autoencoder-based learning solution, we summarize all possible attention patterns that individual heads could learn. Examining the attention strengths and patterns of the important heads, we answer *why* they are important. Through concrete case studies with experienced deep learning experts on multiple ViTs, we validate the effectiveness of our solution that deepens the understanding of ViTs from *head importance*, *head attention strength*, and *head attention pattern*.**

***Index Terms*—Deep learning, explainable artificial intelligence, multi-head self-attention, vision transformer, visual analytics.**

I. Introduction

TRANSFORMER models have demonstrated outstanding performance on tasks in natural language processings (NLP) [1], [2], [3] and time-series forecastings [4]. Recently, their success has also been extended to the vision domain, and the resulting vision transformer (ViT) has achieved on-par and even better performance than the state-of-the-art CNNs [5]. ViT converts a 2D image into a 1D sequence by decomposing it into many patches and arranging the patches sequentially. Each patch is analogous to a token of sequential data, and the multi-head self-attentions are then applied to the sequence to learn the relation among tokens.

Despite the superb performance, it remains unclear how ViT works internally, especially how the multi-head self-attention works on image patches. To be specific, we found ViT designers are often puzzled by the following questions:

- *First*, how important are individual heads, and is their importance consistent across images? As different heads emphasize different pair-wise attentions, their contributions to the prediction are also different. Identifying the important ones would limit the scope of model analysis.
- *Second*, how strong is the attention between two patches that are nearby or far away from each other, and does the attention strength show any trend across layers? It is widely known that CNNs extract basic shapes/colors in early layers but complex objects/concepts in later layers. Since ViTs demonstrate on-par performance with CNNs, it becomes a natural question to ask if the models have any learning heuristic from early to later layers.
- *Third*, what attention patterns have individual heads learned, and are those patterns related to image contents? We have observed heads with interesting patterns, e.g., always attending to the patch itself regardless of the image content (content-agnostic) or only attending to patches with target objects (content-relevant). But, there lacks an exhaustive summary of all possible patterns.

Answering these questions will provide a fundamental understanding of ViTs and assist their further development.

However, there are multiple challenges in answering them. *First*, Michel and Levy [6] proved that heads are not equally important in transformers through intensive ablation studies. Hao et al. [7] proposed a self-attention attribution score for each head to quantify its importance through token interactions. These works focus on NLP tasks and use a head's impact on the prediction to verify its importance. In ViTs, however, we find that heads with little impact on the ultimate predictions may considerably influence the intermediate representations, indicating the need to profile head importance from multiple perspectives. *Second*, for attention strengths and attention patterns, multiple works have proposed to visualize them with heatmap, flow map, or matrix visualizations [8], [9], [10], [11]. However, all these works focus on language transformers with 1D attentions (forward/backward). ViTs, though rearrange image patches into 1D,

Manuscript received 26 October 2022; revised 13 January 2023; accepted 14 February 2023. Date of publication 27 March 2023; date of current version 8 May 2023. This work was supported in part by the National Institute of Health under Grants 1R01CA270454-01 and 1R01CA273058-01. Recommended for acceptance by J. Choo, T. Ropinski, and Y. Hu. *(Corresponding author: Yiran Li.)*

Yiran Li and Kwan-Liu Ma are with the University of California, Davis, CA 95616 USA (e-mail: ranli@ucdavis.edu; klma@ucdavis.edu).

Junpeng Wang, Xin Dai, Liang Wang, Chin-Chia Michael Yeh, Yan Zheng, and Wei Zhang are with the Visa Research, Palo Alto, CA 94301 USA (e-mail: junpenwa@visa.com; xidai@visa.com; liawang@visa.com; miyeh@visa.com; yazheng@visa.com; wzhan@visa.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2023.3261935, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2023.3261935

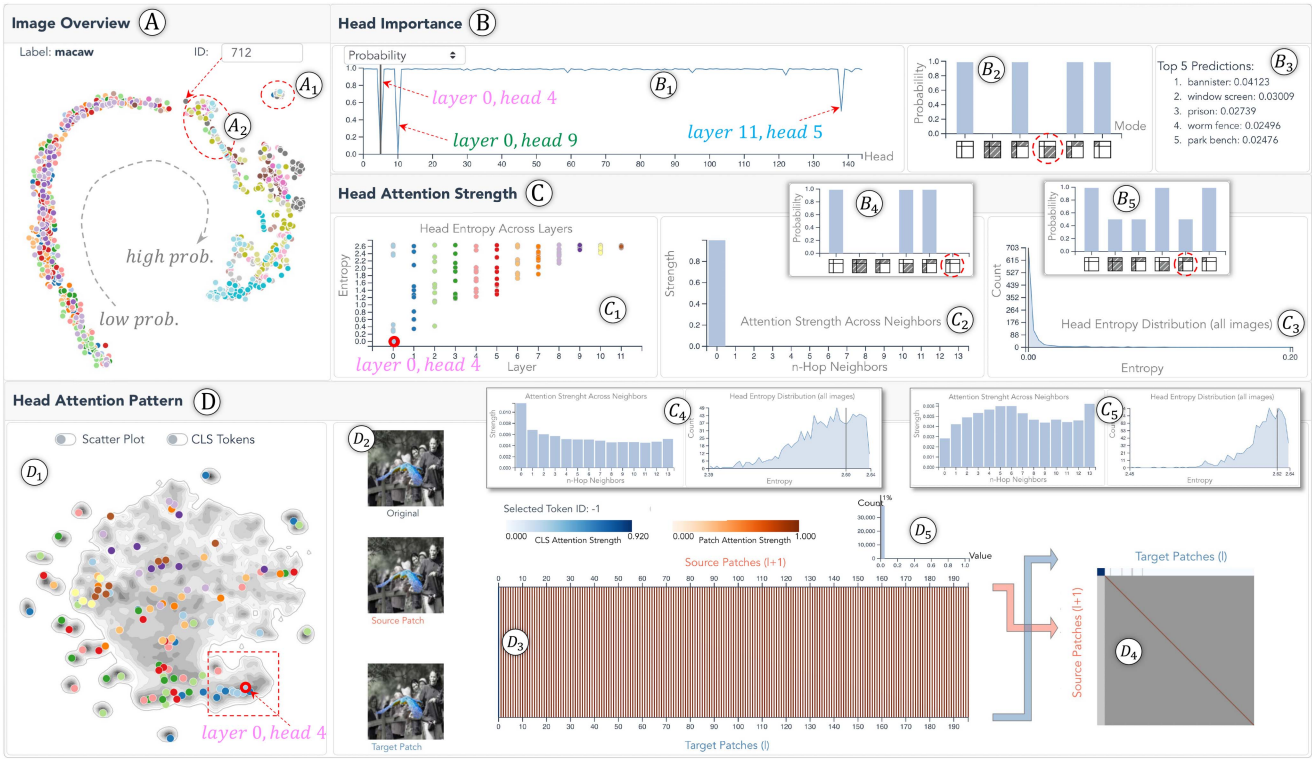

Fig. 1. Our system contains four components. The *Image Overview* (A) lays out all images for selection. The *Head Importance View* (B) shows all heads in different importance metrics (B1) and dissects a head's importance through partial pruning (B2, B3). The attention strengths between patches in a head are shown in the *Attention Strength View* (C), where users first obtain an overview of all heads (C1), and then focus on one head (C2, C3). The *Attention Pattern View* (D) clusters all heads by their attention pattern (D1), and presents the pattern details in one head (D2∼D5).

still present 2D attention behaviors in the 2D spatial context (e.g., upward/downward attention). The 2D attention behavior results in much richer attention patterns, and the patches' attention strength to their neighbors needs to be redefined considering the spatial distance in 2D. *Lastly*, based on our collaborations with domain experts, existing ViT analyses are often piece-by-piece and lack a fluent analytical workflow. For example, to understand the attention between image patches, the patches and attentions between them need to be presented intuitively and explored coordinately. However, most ViT researchers still connect the two parts manually by eyeballing them back and forth.

Our work interprets ViTs from three aspects: *head importance*, *head attention strengths*, and *head attention patterns*. For head importance, we introduce multiple pruning-based head importance metrics, which are computed offline and can be easily plugged into our system to support head exploration and importance analysis. For attention strengths, we profile a patch's attention strength to its k-hop neighbors as a k-dimensional vector. Aggregating the vectors from all patches of a head reflects the head's attention strength distribution. For attention patterns, we train an autoencoder for the heads' attention matrices, and summarize all possible attention patterns by clustering the latent representations of all heads. The three parts are integrated into a coordinated visual analytics (VA) system. We validate the system's efficacy by studying different ViTs with experienced deep learning experts. In short, our contributions include:

1) Multiple pruning-based metrics describing ViT heads' importance from different perspectives.
2) A characterization of heads' attention strength across image patches' k-hop neighbors.
3) A comprehensive summary of the possible attention patterns in ViTs using an autoencoder-based solution.
4) An interactive visual analytics system integrating the above three parts for coordinated interpretations of ViTs.

II. Related Work

Our work belongs to the visual analytics attempts towards more interpretable deep learning (DL), with a special focus on interpreting multi-head self-attention from transformers. We thus review earlier works from these two aspects.

Visualizations for DL. A plethora of visualization works have been introduced for the interpretation of deep neural networks recently [12], [13], [14], [15], [16]. We refer readers to recent surveys [17], [18] for a thorough review of these works. Lately, deep transformers demonstrate superior performance than other DL models on 1D sequential data, and multiple visualization works have been introduced for their interpretations [8], [9], [10], [11], [19], [20]. The success of transformers has also been extended to 2D images with the seminal work of vision transformers (ViTs) [5]. However, to the best of our knowledge, no comprehensive visual analyses have been conducted to demystify this type of powerful yet complex models, especially

how attention works in the 2D image context. Our work tries to fill this gap.

Attention Visualization. The attention mechanism [21] has been used extensively in DL, especially NLP-related tasks, to learn what target tokens the source tokens should "look at". Essentially, attention is a matrix where each cell denotes the attention magnitude that the source token (row) pays to the target (column). Popular attention visualization techniques include flow maps [22], [23], parallel coordinates plots (PCPs) [19], and heatmaps [9], [20], [24]. For example, the flow maps used by Dong et al. [22] connect the source and target tokens with curves, the widths of which denote the attention strengths. Vig [19] arranges the source and target tokens along two parallel axes (i.e., a simplified PCP) and connects them with line segments in between to show the attention patterns. Heatmaps are used extensively in NLP [8], [11], [25], where the attention strengths are directly encoded into the color of each heatmap cell. There are also customized visual designs for attention visualizations [10], [20]. For instance, DeRose et al. [10] extract an "attention graph" from the attentions across layers of a BERT model and arrange the graph into a radial layout to present the propagation of attentions layer-by-layer. The resulting visualization, named Attention Flows, helps to easily analyze and compare attentions from two transformer models.

For *attention patterns* (in individual transformer heads), researchers have discovered some typical ones [9], analyzed their occurrence in different tasks [8], compared the patterns between low and high-performing models [26], and related them with the corresponding heads' importance [11]. However, these works all focus on 1D sequential data. Attentions learned from images with a 2D spatial context have much richer patterns that are difficult to be identified and summarized manually. Our work intends to efficiently discover them. For *attention strengths* between patches within a ViT head, *mean attention distance* has been introduced in previous works [5], [27], [28], which is a sum of the attentions between patches weighted by their spatial distance. This single aggregated value holistically reflects each head's attention strength, but also averages out many spatial details. Here, we introduce the *attention strength vector* to comprehensively profile the spatial distribution of attention strengths.

III. Background

ViT Model. The most popular ViT application is image classification, which is also the focus of this paper. As shown in Fig. 2①–⑤, a ViT classifier runs in five key steps:

1) *Decompose the input image into a sequence of patch tokens*. Without loss of generality, we assume the same width and height for each input RGB image, denoted as w. If the patch size is $pz \times pz$, the number of patches will be $p^2 = \frac{w}{pz} \times \frac{w}{pz}$. The patches are then arranged into a sequence of tokens; each is encoded as an h-dimensional (hD) vector. Each patch token learns a concise representation for the corresponding image patch.
2) *Concatenate* CLS. A zero-initialized hD class token (CLS) is concatenated with the p^2 patch tokens, resulting in a $(1+p^2)\times h$ matrix. CLS learns to accumulate class-related features used to generate the final class probability.
3) *Add positional encodings.* The zero-initialized positional encodings are added to the $(1+p^2)\times h$ matrix. They are trained to learn each patch's positional information. We skip their details as they are not our interpretation focus.
4) *Multi-head self-attention.* This step contains l stacked attention layers, each with n attention heads. Each head learns a $(1+p^2)\times(1+p^2)$ attention weight matrix A, reflecting the pair-wise attention between all $1+p^2$ tokens.
5) *Use the* CLS*token for prediction.* This step decouples the CLS embedding from the patch tokens, and transforms it into class logits through fully-connected layers.

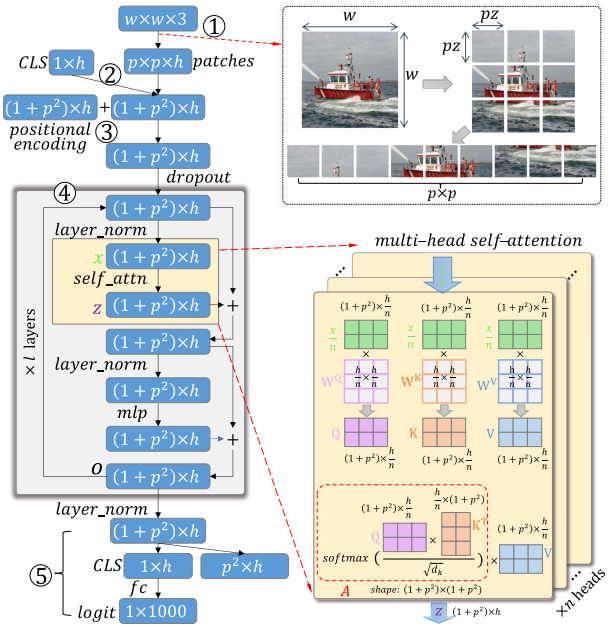

Fig. 2. ViT executes in five steps: (1) decompose the input image into patch tokens; (2) concatenate the CLS token; (3) add the positional encoding; (4) multi-layer multi-head self-attention; (5) use CLS for prediction. Step (4) includes l attention layers, each has n heads. The attention weight in each head, i.e., A, is our interpretation focus.

Self-Attention. Step 4 is the most important. The self-attention in each attention head (one yellow slice in Fig. 2, right) gathers information from all $1+p^2$ tokens to learn how much attention each token should pay to itself and others. The attentions are then used to update the tokens' representations. Specifically, the $(1+p^2)\times h$ matrix at the end of Step 3, after some dropout and normalization layers, is evenly split over the n heads, each with the shape of $(1+p^2)\times\frac{h}{n}$. Inside each head, the matrix is further transformed into Q, K, and V through three separate learnable weight matrices W^Q, W^K, and W^V. The self-attention is then computed as:

$$Attention(Q,K,V) = A{\cdot}V = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right){\cdot}V. \quad (1)$$

The attention weight A of size $(1+p^2)\times(1+p^2)$ encodes the pair-wise attention between all $1+p^2$ tokens. For clarity, we call

the $1+p^2$ tokens *source* tokens when they attend to others, but *target* tokens when they are attended by others.

Multi-Layer and Multi-Head. The self-attention computation is conducted in all n heads, and the resulting attentions are concatenated and linearly transformed to generate the final multi-head self-attention, denoted as z, i.e.,

$$z = Concat(head_1, head_2, \ldots, head_n) \cdot W^O + b, \quad (2)$$

where $head_i = Attention(Q_i, K_i, V_i)$. As shown in Fig. 2④, z will go through more layers to generate the final attention layer output o with shape $(1+p^2)\times h$, which is the updated hD representation for the $1+p^2$ tokens. It will be fed to a new self-attention layer, and the process is repeated for l times, resulting in l stacked layers. In total there are $l\times n$ attention heads, each with an attention weight matrix A recording the learned attention between patches in the respective heads.

IV. Requirements and Solution Overview

We maintained weekly discussions with five domain experts (all are full-time researchers with 5+ years of deep learning experience) working on transformers in vision, NLP, and time-series domains. Over these discussions and our review of the tasks in related literature [6], [7], [9], [10], we elicit the following design requirements for a visual analytics system.

R1: *Head Importance.* To start the interpretation, we first need to quantify the importance of a large number of heads and dissect their importance. Specifically, this requires us to:

- *R1.1*: Assess the importance of a ViT head. We want to reflect a head's impact on both its own attention layer and the ViT's final predictions. The impact should be assessed both on a single image and over all images.
- *R1.2*: Dissect a head's importance. This is to disclose the contributions from two types of tokens to a head's importance: the CLS learns class-related features for prediction; the patch tokens learn important image contents.
- *R1.3*: Use head importance to guide image explorations. To analyze important heads, users need to select the right images for which the corresponding heads show importance. Therefore, we need to provide an informative overview of a large number of images based on their head importance to guide their exploration.

R2: *Head Attention Strength.* From the original ViT paper [5] and the domain experts, we noticed that most ViT designers are wondering how the patches distribute their attention strengths spatially, e.g., whether they attend more to near/far patches and how the attention strength distribution is different across heads. Thus, we should answer:

- *R2.1*: For a single head of an image, how strong are patches attending to their spatially near/far neighbors?
- *R2.2*: For all heads of an image, does their attention show any patterns across layers? What are the patterns?
- *R2.3*: For a single head, does its attention strength show consistent spatial distributions across all images?

R3: *Head Attention Pattern.* As ViT shows more and much richer attention patterns in the 2D context, it is crucial to disclose them with the image semantics. Thus, we need to:

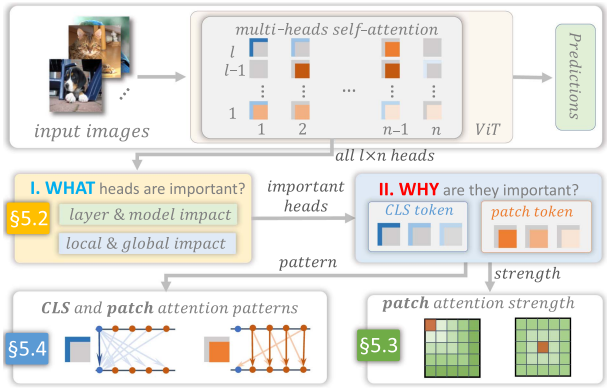

Fig. 3. Overview of our visual interpretation solution for ViT.

- *R3.1*: Exhaustively summarize all possible attention patterns from the $l\times n$ heads (for both CLS and patch tokens) and provide an effective overview of the patterns.
- *R3.2*: Drill down to individual heads of an image to effectively present its attention pattern and investigate if it is agnostic/relevant to the image contents.

Solution Overview. We design a visual analytics system to meet the above requirements. Fig. 3 illustrates the system's workflow. To interpret a well-trained ViT, we first feed the image of interest into the model to get its $l\times n$ heads. Next, we answer *what* heads are important (Fig. 3(I)) through four pruning-based metrics, meeting R1. Focusing on these important heads, we explain *why* they are important (Fig. 3(II)) from two perspectives. First, we disclose the attention strength distribution in individual heads by averaging attention strength across k-hop neighbors of individual patches (R2). Second, using an unsupervised clustering method, we summarize the attention patterns in both CLS and patch tokens and visualize the patterns in important heads (R3). An integrated visualization system (Fig. 1) has been developed following the workflow.

V. Methodology and Visualization System

Our visual analytics system (Fig. 1) contains four components: the *Image Overview*, the *Head Importance View*, the *Attention Strength View*, and the *Attention Pattern View*. The *Image Overview* (Fig. 1(A)) lays out image instances based on their heads' importance vector, providing an entry point for the exploration. The remaining three views (Fig. 1(B)–(D)) are designed to meet the three requirements.

A. The Image Overview

The *Image Overview* (Fig. 1(A)) uses tSNE+scatterplot to provide an overview of the images. Each point represents an image, and its color denotes the class label. The coordinates of each point are the dimensionality reduction result of the corresponding image's *head importance vector*, i.e., a $(l\times n)$-dimensional vector with the corresponding head's importance (3) at each dimension. The tSNE layout based on this vector clusters images with similar head importance together, guiding users' exploration (*R1.3*). For example, there is a small cluster in

the top-right corner, which immediately catches users' attention during exploration (see Section VI).

Clicking on any point or providing an ID in the top-right input box will select the corresponding image into the other three views. Inside each view, the analysis can focus on the selected image or be extended to all other images.

B. The Head Importance View

We define several metrics to quantify the importance of a head. These metrics are generated by "leave-one-out" ablations, i.e., encoding a head's importance by the changes in the final output (*model-level impact*) or next-layer activations (*layer-level impact*) after pruning the head. Pruning a head is conducted by setting its attention matrix (A in (1)) to 0. Similar head/neuron importance analysis through ablation studies has been widely adopted in NLP, e.g., [6], [7], [29].

*1) Importance to the Model's Output (*R1.1*):* We propose two *model-level* importance metrics for each head. One reflects the probability change of the true class (3); the other encodes the Jensen-Shannon Divergence (JSD) between the two probability distributions (4) before and after a head is pruned. Mathematically, $ViT()$ denotes the well-trained model, which takes an image as input and outputs a probability distribution, i.e., $\mathbf{P} = ViT(img)$. $ViT_{i,j}()$ is the same model but the jth head from the ith layer has been pruned, and $\mathbf{P}_{i,j} = ViT_{i,j}(img)$. idx_{label} is the image's true class index. The importance of head (i, j) is:

$$I_{i,j}^{prob} = \mathbf{P}[idx_{label}] - \mathbf{P}_{i,j}[idx_{label}] \tag{3}$$

$$I_{i,j}^{JSD} = JSD(\mathbf{P}||\mathbf{P}_{i,j}) \tag{4}$$

*2) Importance to the Attention Layer (*R1.1*):* Assessing only the changes in final outputs cannot reflect a head's importance in its attention layer. As our experts noticed, pruning an important head may significantly change the corresponding layer's output (i.e., z in (2)), but show minor changes to the final probabilities. This is because heads from later layers may compensate for the contribution of the pruned head, concealing its importance.

To identify the important heads in each attention layer, we propose two *layer-level* importance metrics, which are defined by the cosine distance (D_{cos}) between the immediate layer activations before and after a head is pruned. As shown in Fig. 2, the attention layer's output z is a $(1+p^2)\times h$ matrix, containing the activations of the CLS (the first $1\times h$) and patch tokens (the later $p^2\times h$). Our layer-level metrics measure the importance of the CLS and patch tokens separately. For CLS, the metric ($I_{i,j}^{\texttt{CLS}}$) reflects the cosine distance between the two CLS activations. For patch tokens, the metric ($I_{i,j}^{patch}$) similarly computes the cosine distances and averages the distances over all patches. Mathematically (z and z' are the layer's output before and after pruning),

$$I_{i,j}^{CLS} = D_{cos}(z[0], z'[0]) \tag{5}$$

$$I_{i,j}^{patch} = \frac{1}{p^2}\Sigma_{i=1}^{p^2} D_{cos}(z[i], z'[i]) \tag{6}$$

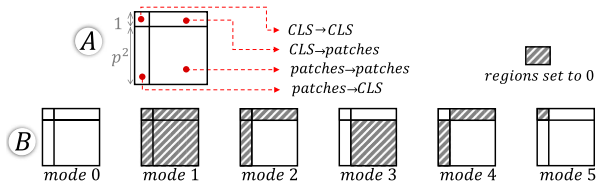

Fig. 4. Pruning modes. (A) The attention matrix is divided into four regions. (B) Different pruning modes set one/multiple regions to 0.

*3) Head Pruning Modes (*R1.2*):* Once the important heads are identified, we further dissect their importance by partially pruning them.

As shown in Fig. 4(A), the attention matrix of a head can be divided into four regions based on the source and target tokens: CLS→CLS, CLS→*patches*, *patches*→CLS, and *patches*→*patches*. Regions CLS→*patches* and *patches*→CLS are considered together, as they both encode the interaction between the CLS and patch tokens. Six pruning modes are defined by setting different regions to zero (Fig. 4(B)), i.e., mode 0 is the original head without pruning; mode 1 prunes the head completely; modes 2∼5 are additional cases where only the striped regions are pruned. Showing the impacts from these modes attributes the head's importance to individual regions.

4) Visualization: The *Head Importance View* (Fig. 1(B)) visualizes our proposed metrics and pruning modes. First, Fig. 1(B1) uses a line chart to present the four head importance metrics for a single selected image (i.e., the heads' *local* importance to an image). The horizontal axis represents all the $l\times n$ heads, and the vertical axis denotes a metric's value, where the dropdown widget enables users to switch among the four metrics. Note that for $I_{i,j}^{prob}$, we directly show the value of $\mathbf{P}_{i,j}[idx_{label}]$ (instead of the difference in (3)) as it is more intuitive. When no image is selected (e.g., at the beginning of exploration), the curve in this view shows the average value of the selected metric over all images. Meanwhile, a blue band surrounding the curve denotes the standard deviation of the metric's values (see Fig. 7). The mean and standard deviation reflect the *global* importance over all images, guiding users to select globally important heads.

Second, after a head is selected from Fig. 1(B1) (by dragging the vertical line), the bar-chart in Fig. 1(B2) shows the selected importance metric (y-axis) in different pruning modes (x-axis), further dissecting the head's importance. For example, Fig. 1(B2) reveals that the importance of the selected head originates from the patch tokens solely, and pruning CLS-related attentions shows no impact.

Lastly, Fig. 1(B3) shows the top-5 predicted probabilities for the selected image, in the current pruning. If the true label is among the top 5, it will be highlighted in bold.

C. The Attention Strength View

The attention strength of a head characterizes the spatial distributions of the attention strength across all patches, which

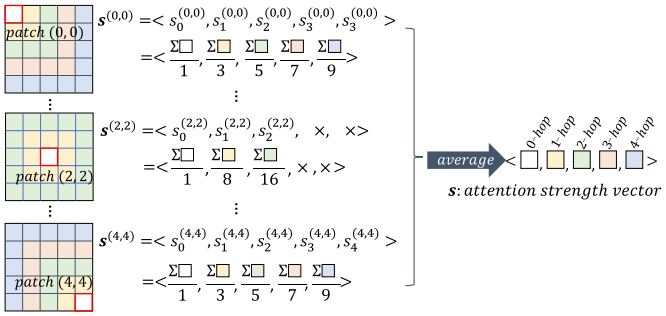

Fig. 5. Computing the attention strength vector for a single head.

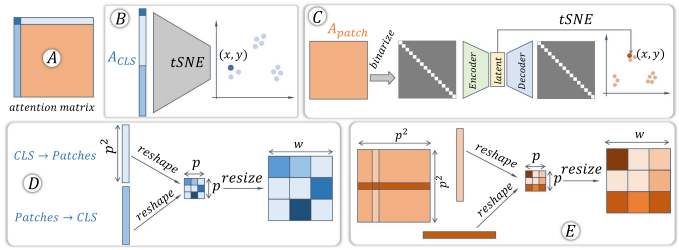

Fig. 6. (A) Each attention matrix is separated into CLS-related attentions ($A_{\texttt{CLS}}$, blue) and patch attentions ($A_{\texttt{patch}}$, orange). (B) $A_{\texttt{CLS}}$ from all images is visualized through tSNE+scatterplot. (C) For $A_{\texttt{patch}}$ from all images, we train an autoencoder to cluster them. (D, E) $A_{\texttt{CLS}}$ and $A_{\texttt{patch}}$ can be mapped back to the image as a mask.

answers why the head is important by disclosing where it makes the patches focus.

*1) Attention Strength Over k-Hop Neighbors (*R2.1*):* We define a p-dimensional (pD) *attention strength vector*, $\mathbf{s}$, for each head, which profiles the average attention strength of all patches to their k-hop neighbors ($k \in [0, p-1]$), i.e.,

$$\mathbf{s} = \frac{1}{p^2}\Sigma_i\Sigma_j \mathbf{s}^{(i,j)},\ i\in[0,p-1],\ j\in[0,p-1], \tag{7}$$

$$\mathbf{s}^{(i,j)} = < s_0^{(i,j)}, s_1^{(i,j)}, s_2^{(i,j)}, \ldots, s_{p-1}^{(i,j)} >, \tag{8}$$

where $s_k^{(i,j)}$ denotes the average attention from patch (i, j) to its kth hop neighbors in the 2D domain.

Without loss of generality, Fig. 5 shows the computation of $\mathbf{s}$ when $p = 5$. Starting from patch (0,0), $s_0^{(0,0)}$ is the attention that patch (0,0) paid to itself (i.e., 0-hop attention); $s_1^{(0,0)}$ is the sum of the attentions paid to its 1-hop neighbors in yellow divided by the number of neighbors (i.e., 3); $s_2^{(0,0)}$ is the total attentions paid to its 2-hop neighbors in green divided by the number of neighbors (i.e., 5); and so on so forth. To the end, we get a 5D vector for patch (0, 0), i.e., $\mathbf{s}^{(0,0)}$. Repeating this computation to all patches, we get 25 5D attention strength vectors, one for each patch. Their average is the head's *attention strength vector*, i.e., $\mathbf{s}$.

Note that some patches may not have certain hops of neighbors, e.g., patch (2,2) in Fig. 5 does not have 3-hop or 4-hop neighbors (marked as '×'). Therefore, $s_3^{(2,2)}$ and $s_4^{(2,2)}$ will not be counted when computing the corresponding element of vector $\mathbf{s}$. In other words, the denominator in (7) is not p^2 for all elements of $\mathbf{s}$; some will have a smaller denominator due to the missing neighbors.

2) Visualization: The *Attention Strength View* (Fig. 1(C)) presents all heads' attention strength with three components. The first component (Fig. 1(C1)) presents an overview of all heads for the selected image through a scatterplot. Each point in the scatterplot is a head. Its horizontal position (as well as its color) reflects the layer that the head is from. Its vertical position denotes the entropy of the head's attention strength vector $\mathbf{s}$ (normalized). The entropy of $\mathbf{s}$ reflects if the head's attention strength is localized on a certain-hop of neighbors (low-entropy, one element's value dominates the vector) or spread across all k-hop neighbors (high-entropy, all elements' values are similar). From the overview, there is an obvious trend of the heads across layers (*R2.2*), i.e., heads from higher layers attend more evenly to all patches, whereas lower-layer heads attend either locally or globally.

Second, after a head is selected by clicking the corresponding point in Fig. 1(C1), its attention strength vector is presented as a bar chart in Fig. 1(C2) (*R2.1*). In the current visualization, we can see that all patches in the selected head attend only to themselves (i.e., all attention strengths are distributed to the 0-hop neighbors).

Lastly, the area plot in Fig. 1(C3) presents the distribution of entropy values for the selected head over all images (*R2.3*). For example, the currently selected head has a small entropy (Fig. 1(C1)) as all patches attend to 0-hop neighbors only (Fig. 1(C2)). Fig. 1(C3) further reveals that the head has consistently low entropy across all images, reflected by the peak on the left corner. The vertical line over the area plot marks the head's entropy for the currently selected image, reflecting how much the head's attention strength for the current image varies from its strength for other images.

D. The Attention Pattern View

The attention pattern of a head reflects how tokens are attending to each other. We want to summarize the possible patterns of all heads to deepen the understanding of ViTs.

*1) Unsupervised Pattern Identification (*R3.1*):* Due to the functionality difference between the CLS and patch tokens, we treat them separately and learn their respective patterns. Specifically, given an input image and one of its heads, the corresponding attention matrix A is of shape $(1+p^2)\times(1+p^2)$ (Fig. 2). We separate A into CLS-related attentions $A_{\texttt{CLS}} = concat(A[0,:], A[1:,0]) \in \mathbb{R}^{2p^2+1}$ and patch attentions $A_{patch} = A[1:,1:] \in \mathbb{R}^{p^2\times p^2}$, as shown in Fig. 6(A).

CLS Attention Patterns. The CLS-related attentions ($A_{\texttt{cls}}$) concatenates the CLS→CLS, CLS→*patches*, and *patches*→CLS regions (Fig. 6(A)), and its size is $(2p^2+1)$. If we have m images, each generates $l\times n$ attention matrices from the $l\times n$ heads, we will have $m\times l\times n$ such vectors. Using tSNE, we project them from $(2p^2+1)$D to 2D and present them with a scatterplot (Fig. 6(B)). Attention heads with similar CLS attention patterns will be clustered together.

Patch Attention Patterns. We applied the same method to the patch attentions $A_{patch} = A[1:,1:] \in \mathbb{R}^{p^2 \times p^2}$, but the resulting tSNE layout could not clearly separate/cluster dissimilar/similar patch attention patterns. We believe this is caused by the much higher dimensionality of the patch attentions and tried to fix it with several remedies. For example, we used max pooling to spatially shrink A_{patch} before tSNE, and tried to apply PCA on A_{patch} before tSNE. Both solutions did not yield much performance gain.

In the end, we came up with an autoencoder (AE)-based learning solution (Fig. 6(C)). *First*, as we care more about the attention pattern, instead of the magnitude, we binarize A_{patch} using a cutoff, e.g., setting top 1% values to 1 and the rest to 0. This enhances the patterns and makes them easier to learn. *Second*, using the $m \times l \times n$ binarized A_{patch}, we train an AE. The AE has two symmetric subnetworks, i.e., the encoder and decoder, each with two convolutional layers and one fully-connected layer. *Third*, using the latent representations from the well-trained AE's bottleneck layer, we conduct tSNE layout. The layout shows obvious clusters, exposing different attention patterns.

2) Visualization: The *Attention Pattern View* adopts the "overview+details" exploration strategy to visualize the attention patterns.

The *overview* presents all heads from all images ($m \times l \times n$ in total) through tSNE+scatterplot (*R3.1*, Fig. 1(D1)), as explained in Section V-D1. The tSNE layout could be either for the CLS attentions ($A_{\texttt{CLS}}$) or for the patch attentions (A_{patch}). The top-right toggle enables this switch. To be scalable, we allow users to convert the scatterplot into a density plot, and the top-left toggle controls this. For example, the background density contours in Fig. 1(D1) present the distribution of all the $m \times l \times n$ heads, as a context. When an image of interest is selected from the *Image Overview* (Fig. 1(A)), its $l \times n$ heads will be shown on top of the density plot as points, the color of each reflects its layer.

The *details* of the attention matrix for a selected head are shown in the right of Fig. 1(D) (*R3.2*). An attention matrix denotes the attention between $(1+p^2)$ tokens, and we present it in two different manners (Fig. 1(D3) and (D4)).

Fig. 1(D3) lists all $(1+p^2)$ tokens as two rows (top: source tokens, bottom: target tokens) and uses lines with light to dark color to encode the attention magnitude. Blue and orange are used to color $A_{\texttt{CLS}}$ and A_{patch} respectively. Showing all the $(1+p^2) \times (1+p^2)$ lines would make the view very cluttered. Therefore, we enable users to specify a threshold, the lines with associated attention value below which will be disabled. The histogram in Fig. 1(D5) shows the distribution of the $(1+p^2) \times (1+p^2)$ values, guiding users to specify the threshold by dragging the vertical bar on top of the histogram. The current threshold in Fig. 1(D5) is 1%, indicating only the top 1% lines are visible. From the dark vertical lines in Fig. 1(D3), we can easily see that all tokens (both CLS and patch tokens) strongly attend to themselves.

Fig. 1(D4) presents the attention matrix through a heatmap (row: source token; column: target token). The four parts of the heatmap have been illustrated in Fig. 4(A). For A_{patch} in the bottom-right corner, one pixel represents one attention value. For $A_{\texttt{CLS}}$, as one pixel is barely visible for the single row and column of attention values, we augment them to take 10 pixels. The color mapping is consistent with that in Fig. 1(D3) (blue: $A_{\texttt{CLS}}$; orange: A_{patch}). From the heatmap, we can observe a clear diagonal pattern, indicating that the patch tokens attend strongly to themselves. The CLS token also strongly attends to itself, as the top-left cell is in dark blue. Meanwhile, CLS also attends to different patch tokens, but the attention magnitude is very small (light blue or white color in the top row). The threshold specified from the histogram in Fig. 1(D5) also applies to this heatmap.

The reason for presenting the attention matrix in two visualizations is to leverage their respective advantages. The heatmap shows the attention patterns more intuitively, whereas the two-axes view can better present the attention relationship between the CLS and patch tokens (one example is shown later in Fig. 14). The two-axes view is also better than the heatmap in terms of interacting with tokens, e.g., it can easily highlight a token of interest (see Fig. 11). Apart from these two, we have also considered other visualizations in our early design stages. For example, we tried to overlay arrows on top of the heatmap to show the attention direction or embed patch pixels into the heatmap. However, both designs are not easily scalable to our problem size.

Image Context. To intuitively present the patch-related attentions, we need to map the patches back onto the 2D image. For CLS$\rightarrow$*patches* (shape: $1 \times p^2$) and *patches*$\rightarrow$CLS (shape: $p^2 \times 1$) attentions, we reshape them to a $p \times p$ square, scale the square to $w \times w$, and overlay it on top of the image as a transparency mask (Fig. 6(D), stronger attention $\rightarrow$ more transparent). For the *patches*$\rightarrow$*patches* attentions (shape: $p^2 \times p^2$), we reshape individual row/column into a $p \times p$ square, scale it to $w \times w$, and overlay it on top of the image as a mask to show the attention from a token to all others (a row) or vise versa (a column), Fig. 6(E). The three images in Fig. 1(D2) (top-bottom) show the original image, image+source attention mask, and image+target attention mask. Hovering over individual source/target tokens from the top/bottom axis in Fig. 1(D3) will update the source/target attention masks dynamically (e.g., Figs. 10(D) and 11(C)).

VI. Case Study and Experts' Feedback

We use multiple case studies, conducted together with deep learning experts, to show the capability of our system. The experts' feedback is summarized at the end of Section VI-E.

For ViTs, we explored four pre-trained ViTs with different image resolutions (w), numbers of layers (l) and heads (n) [5]. As our findings are consistent across them,we focus our illustrations on one model only but include results of the other three in our Appendix, available online. The parameters of the focused model are: $w = 224$, $l = 12$, $n = 12$, and $p = 14$.

For datasets, we used 1000 images sampled from the validation set of ImageNet [30]. The images are from 20 classes (10 classes with the best and 10 classes with the worst predictions), each has 50 images. We have also explored our system with

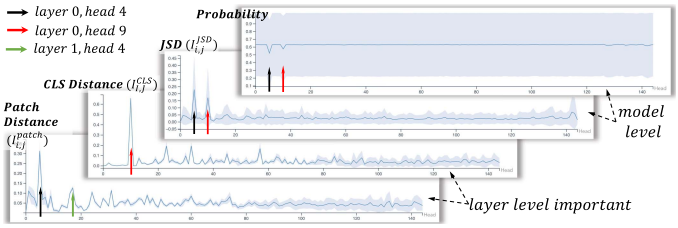

Fig. 7. The mean (blue curve) and standard deviation (blue band) of the four head importance metrics computed over all images.

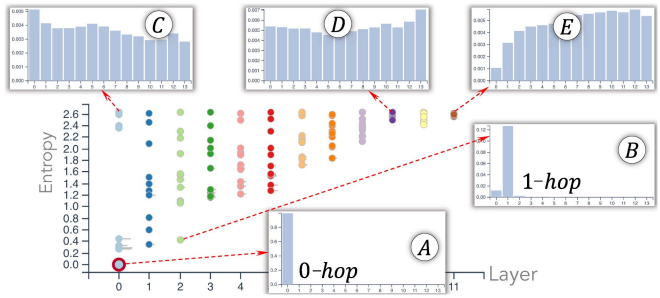

Fig. 8. Lower-layer heads can distribute patches' attention strength only to their near neighbors (A, B), or evenly to all k-hop neighbors (C). Higher-layer heads can only do the latter (D, E).

another dataset. However, as the model-level findings from the two datasets are mostly similar, we include the exploration of the other dataset in our Appendix, available in the online supplemental material.

A. Head Importance

Fig. 1(A) shows the 1000 images, laid out by their $l \times n$-dimensional head importance vector (each dimension is $\mathbf{P}_{i,j}[idx_{label}]$ in (3)). From the layout, we found images are arranged clockwise with an increasing true-class probability.

Fig. 7 shows the four head importance metrics aggregated over all images. Both *model-level* metrics ('Probability,' 'JSD') reflect head 4 and 9 from layer 0 are very important. The very wide band of the 'Probability' plot is due to our choice of images with the best and worst performances. The two *layer-level* metrics ('CLS Distance,' 'Patch Distance') show heads 9 and 4 contribute significantly to the changes of CLS and patch representations, respectively.

Moreover, the *layer-level* metrics show more oscillations, identifying important heads that cannot be identified from the *model-level* metrics. For example, removing layer 1 head 4 barely changes the final predictions but significantly affects the layer activations. As shown in the 'Patch Distance' plot, the mean for this head is large and its standard deviation is small, indicating the head is important to the layer across images. This head has a fixed function of making all patches attend to the patch above themselves (explained later in Fig. 12(F)). The *model-level* metrics cannot identify it as heads from later layers show similar functionalities (i.e., head 3 from layer 2, explained later in Fig. 12(J)), hiding its importance. Also, an increasing standard deviation is observed from the two *layer-level* metrics, indicating that higher-layer heads' importance is more influenced by image contents.

By coordinately exploring the *Image Overview* and *Head Importance View*, we find image clusters, to which, individual heads are very important (e.g., heads 9 and 4 from layer 0 are important to images in Fig. 1(A1) and (A2)). To analyze their importance, we randomly select an image (ID: 712, label: macaw) from one cluster for further exploration. The three heads that are very important to the selected image are: layer 0 head 4, layer 0 head 9, and layer 11 head 5 (Fig. 1(B1)).

Fig. 1(B2) shows the partial pruning results for layer 0 head 4. The probability drops only if the $patches \rightarrow patches$ attentions are pruned, and pruning CLS-related attentions has little impact. This attributes head 4's importance to its attentions between patches. Fig. 1(B3) shows the top-five probabilities are very low when this head is pruned, and macaw (the true label) is not among them.

Fig. 1(B4) and (B5) show the partial pruning results for layer 0 head 9 and layer 11 head 5, respectively. From them, *layer 0 head 9* is important due to the CLS self-attention (CLS→CLS); layer 11 head 5 is important due to the attentions between CLS and patches (CLS→$patches$ and $patches$→CLS).

B. Attention Strength

Next, we examine why the heads are important through their attention strengths. The attention strength analysis is for patch tokens only (CLS has no spatial information), so our analysis focuses on layer 0 head 4. Fig. 1(C1) shows the attention strength overview of all heads. layer 0 head 4 (in the red circle) has the smallest entropy. Fig. 1(C2) shows the head's pD k-hop neighborhood vector. From it, all patches' attention strengths focus on the 0-hop neighbor, i.e., the head makes all patches strongly attend to themselves. It is reasonable that such a head is important in lower layers, as no details should be overlooked at the beginning. From Fig. 1(C3), we also notice that this head's functionality is consistent across all images, as its entropies for different images are always low (distributed dominantly to the left).

Fig. 1(C4) and (C5) show the attention strengths of the other two important heads. Their importance majorly comes from CLS-related attentions, and the attention strengths (for patch tokens) are scattered across all hops of neighbors (the bar chart) and varying across images (the area plot).

The overview in Figs. 1(C1) and 8 also reveals the attention strength distribution of heads over layers. In general, lower-layer heads can make patches attend strongly to their local regions, e.g., 0-hop or 1-hop neighbors (the low entropy heads in Fig. 8(A)–(B)). Low-layer heads can also make patches evenly distribute their attention across the entire image (e.g., Fig. 8(C)). For higher-layer heads, patches only attend globally with similar attention strengths across all k-hop neighbors, e.g., Fig. 8(D)–(E). A similar overview can always be observed no matter which image is selected. This observation is consistent with the original claims about attention strengths [5] (see details

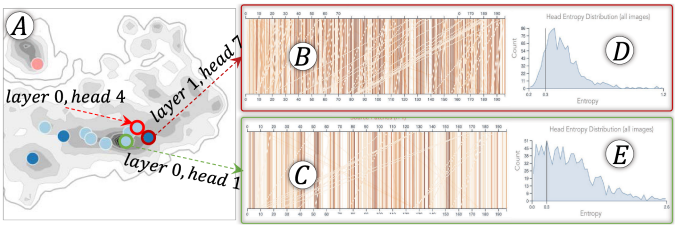

Fig. 9. Exploring heads that have similar attention patterns with layer 0 head 4 to explain why it is important while others are not.

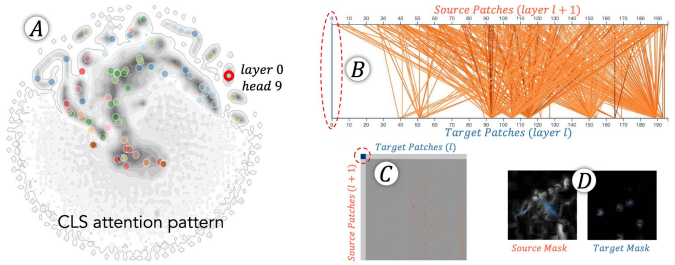

Fig. 10. The attention pattern of layer 0 head 9.

about [5] in our Appendix, available in the online supplemental material).

C. Attention Pattern

The attention patterns further answer why layer 0 head 4 is important. From the *Attention Pattern View*, i.e., the vertical lines in Fig. 1(D3) and the diagonal pattern in Fig. 1(D4), all patches in this head strongly attend to themselves, echoing our earlier findings from the *Attention Strength View*.

Are there heads with attention patterns similar to *head 4*? If yes, why is only *head 4* so important? To answer these questions, we zoom into the red dashed region of the tSNE layout in Fig. 1(D1). The zoomed-in details are shown in Fig. 9(A). From it, we explore heads close to *head 4* and check their attention patterns. Fig. 9(B)–9(C) show two of them, where the vertical lines indicate the patches also majorly attend to themselves in these two heads. However, different from *head 4*, the self-attentions in these two heads are not always strong, i.e., all lines in Fig. 1(D3) are in dark orange, but most lines in Fig. 9(B)–9(C) are in light orange. Moreover, the two heads' functionality is not as consistent as that of *head 4* across images (comparing Figs. 1(C3) and 9(D)–(E)).

For layer 0 head 9, we have known its importance comes from the CLS→CLS attention in Fig. 1(B4). Visualizing its attention patterns, we can see the dark blue vertical line in Fig. 10(B) and the dark blue cell in the top-left corner of Fig. 10(C), echoing the importance of CLS's self-attention. Meanwhile, the CLS→*patches* and *patches*→CLS attentions are not noticeable. The *patches*→*patches* attentions (the orange lines in Fig. 10(B) and the vertical pattern in Fig. 10(C)), as well as the masked source and target images (Fig. 10(D)), show no obvious extracted features, confirming the less importance of the patch-related attentions. Also, we noticed that the CLS attention pattern of *head 9* is very unique, as it is the only head (of the selected image) in the isolated cluster of Fig. 10(A) (the tSNE layout of all heads' $A_{\texttt{CLS}}$). The background contour shows the head 9 from other images, reflecting that head 9's functionality is fixed across images.

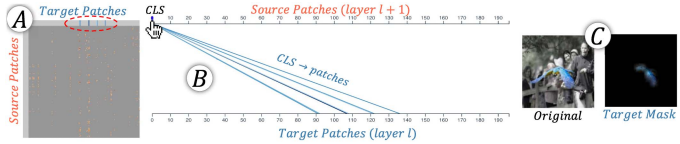

Fig. 11. The CLS→*patches* attention in layer 11 head 5.

For layer 11 head 5, its importance is from the *patches*→CLS and CLS→*patches* attentions (Fig. 1(B5)). In Fig. 11(A), the *patches*→CLS attentions (the leftmost column) are not noticeable, but the CLS→*patches* attentions (the top row) show four major regions, indicating the CLS attends to four groups of patches. Hovering over the CLS token on the top axis of Fig. 11(B), the four groups of target patches are highlighted in the masked target image (Fig. 11(C)). From it, the four regions accurately extract the macaw's wings in blue.

1) Attention Pattern Summary: Our explorations identified different attention patterns. This section provides an exhaustive summary.

Patch Attention (A_{Patch}) Patterns. Fig. 12(A) shows the tSNE layout of all heads' patch attentions from the 1000 images (Fig. 1(D1) is the density plot of it). The points from an isolated cluster often represent the same head from different images, verifying the head's fixed functionality. Exploring individual heads in the *Attention Pattern View*, we summarize them into 13 patterns in Fig. 13. The three rows of the illustrative figure show how each patch attends to others in the image space (top), the two-axes (middle), and the heatmap (bottom). From the last row, each of the 13 patterns is a combination of four basic patterns, i.e., *diagonal*, *horizontal*, *vertical*, and *block*.

Fig. 13(A)–(k) include the *diagonal pattern*. Fig. 13(A) denotes self-attention. Fig. 13(B)–(C) show each patch attends to its left or right patch (one cell off the diagonal), where the gaps indicate the leftmost or rightmost patches without further ones to attend to. Each patch in Fig. 13(D)–(E) attends to the patch above or below itself, i.e., the white-squares shift from the diagonal for a row of patches. The four heads in Fig. 12(C)–(F) (all from layer 1) show examples where each patch attends to its right, bottom, left, and top patch, respectively. The four heads in Fig. 12(G)–(J) (all from layer 2) show similar patterns, revealing the repeating functionalities. Coming to middle attention layers, patches can attend to multiple patches above and/or below themselves (Fig. 13(F)–(H)). Fig. 12(M)–(N) show two such examples from layer 4. Fig. 13(I) includes the counter-diagonal patterns, in which the left/right patches symmetrically attend to the right/left ones in the same row.

Fig. 13(J)–(K) contain the *horizontal pattern* (mixed with the diagonal pattern). The pattern indicates that a patch attends to multiple patches before/after itself in the same row.

Fig. 13(L) shows the *vertical pattern*, i.e., multiple source patches (heatmap rows) attend to the same target patches (heatmap columns). This usually indicates the target patches

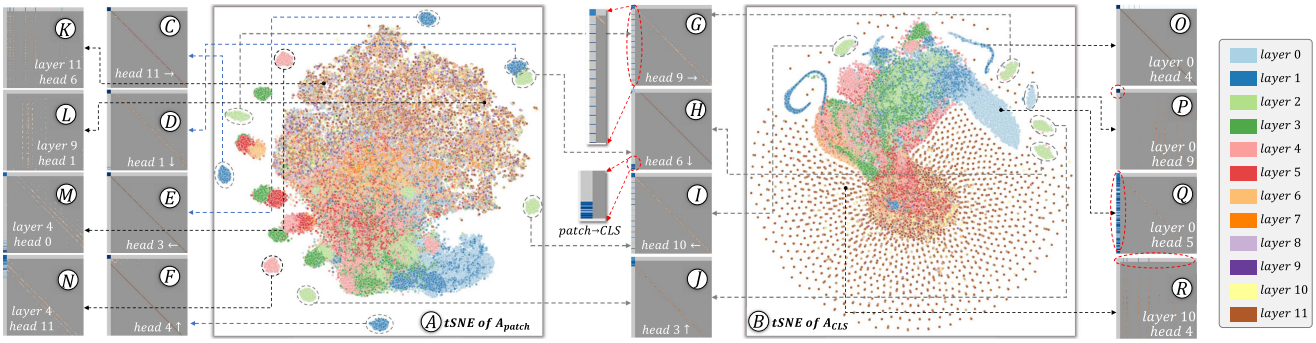

Fig. 12. (A/B) tSNE layouts of heads using their patch/CLS attention patterns. Content-agnostic heads from lower layers show similar patterns across images, and thus each forms an isolated cluster no matter they are laid out by the patch (C-J, M, N) or CLS (G-J, O-Q) patterns. Content-relevant heads from higher layers show dissimilar patterns for different images. These heads are scattered in both layouts (K, L, R).

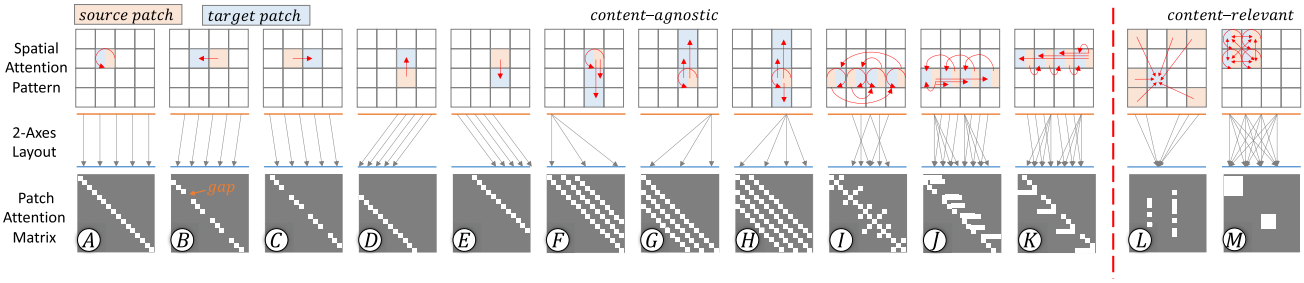

Fig. 13. The 13 possible attention patterns between image patches presented in the (top) image space, (middle) two-axes, and (bottom) heatmap visualization. Each pattern is a mix of one/multiple of the four basic patterns: *diagonal* (A-K), *horizontal* (J, K), *vertical* (L), and *block* (M). The first two are *content-agnostic* and often appear in lower-layer heads; the latter two are *content-relevant* and often occur in higher-layer heads.

include important semantics to the class (e.g., the cat face region of a cat image, Fig. 12(K)–(L)). The pattern often occurs in higher-layer heads, as indicated by the yellow/brown color in Fig. 12(A) (the big chaotic cluster in the center).

Fig. 13(M) shows the *block pattern*, i.e., patches in a local region mutually attend to each other (e.g., the face patches of a cat attend to its body patches and vice versa). Similar to segmentation, the attentions extract the object's pixels.

Attention patterns can also be categorized into *content-agnostic* and *content-relevant*. The *diagonal* and *horizontal* patterns are often agnostic to the image content, e.g., heads in Fig. 12(C)–(J). The same head from all images forms an isolated cluster in Fig. 12(A). The *vertical* and *block* patterns are content-relevant, as their position depends on the content of images. They are the big chaotic cluster in Fig. 12(A). Content-agnostic patterns often occur in lower layers, whereas content-relevant patterns often occur in higher layers.

CLS Attention (A_{CLS}) Patterns. The CLS-related attention patterns follow a similar layer-wise trend. As shown in Fig. 12(B) (its density plot is in Fig. 10(A)), heads in lower layers form clear clusters, indicating the CLS's attentions in them are more content-agnostic. For example, Fig. 12(P) shows dominantly strong CLS→CLS attentions; Fig. 12(Q) shows strong *patches*→CLS attentions. In higher layers, the attentions are more content-relevant, e.g., in Fig. 12(R) (the top row), the CLS focuses only on specific image patches.

The four heads with isolated clusters in Fig. 12(A), (G)–(J) also form four isolated clusters in their CLS layout (Fig. 12(B), (G)–(J)). By coordinately exploring the patch and CLS patterns, we found the boundary patches without further patch to attend in these heads will attend to CLS. For example, in Fig. 12(H), all patches attend to the patch beneath themselves. The bottom row of patches have no patches beneath them, so they attend to CLS (Fig. 14(A)). The two-axes view of this head is shown in Fig. 14(B), which is consistent with the pattern in Fig. 12(H) (and the inset). This explains how information is passed across patches row-by-row to CLS for classification.

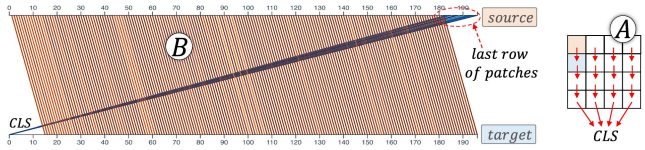

Fig. 14. Each patch attends to the one beneath it, e.g., patch 0 (row 0, column 0) attends to patch 14 (row 1, column 0, $p = 14$). The bottom row of patches all attend to CLS since there is no further patch.

D. Head Attention Diagnosis

Our coordinated system also helps to diagnose the roles of different heads (especially the important ones) in mispredictions. We brief two example cases in this section.

Case 1: Fig. 15(A) shows an image from the overskirt class, but the ViT performs badly on its prediction (Fig. 15(B)). From the *Head Importance View*, we notice a sharp increase

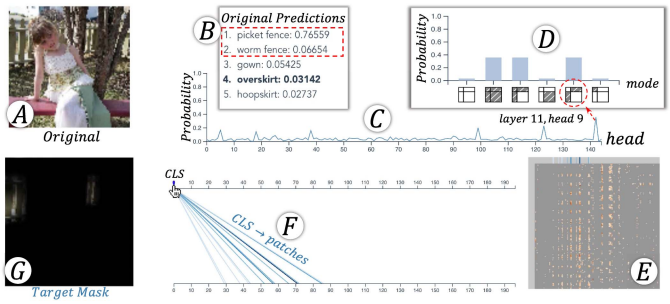

Fig. 15. The head 9 from layer 11 incorrectly attends to the background fence. Pruning it will increase the true class's probability.

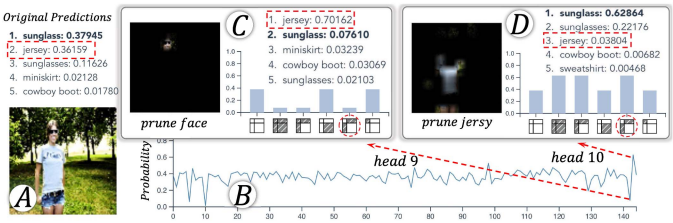

Fig. 16. Heads 9 and 10 (layer 11) extract the `sunglass` and `jersey` features respectively, impacting the corresponding classes' probability.

in the true label probability when head 9 of layer 11 is pruned (Fig. 15(C)). The partial pruning result of this head (Fig. 15(D)) reflects that the probability of `overskirt` will *increase* only if the attentions between `CLS` and patches are pruned. As the *patches*→`CLS` attentions are very small in Fig. 15(E) (the first column), we hover over the `CLS`→*patches* attentions in Fig. 15(F). The masked target image (Fig. 15(G)) shows that the `CLS` attends strongly to the background fences. Pruning such a misleading head will help the model focus more on the right features, leading to a probability increase. The importance of this head also explains why the top two classes in the original predictions (Fig. 15(B)) are `fences`.

Case 2: The ViT correctly predicts the image in Fig. 16(A) to be `sunglass`, but also assigns a high probability to `jersey`. We found two important heads in the last layer but with opposite effects (Fig. 16(B)). Pruning heads 9 and 10 from layer 11 separately leads to a big decrease and increase of the `sunglass` probability. From the partial pruning results, we found the attentions between `CLS` and patches dominate the decrease (Fig. 16(C)) or increase (Fig. 16(D)). We then visualize the `CLS`→*patches* attentions and the corresponding masked images. For head 9, the `CLS` focuses on the face region, whereas for head 10, the `CLS` attends solely to the jersey region. Heads 9 and 10 contribute largely to the probability of `sunglass` and `jersey` respectively. Pruning them increases the opposite class's probability (Fig. 16(C) and (D)). These details significantly deepens the understanding of how ViT works.

E. Domain Experts' Feedback

The above case studies were conducted with 7 deep learning experts in separate sessions, using the protocol of *guided exploration+think-aloud discussions*. All experts are researchers with 5+ years of experience in deep learning. Five experts (E_1∼E_5) have participated in our requirement analysis. The other two (E_6∼E_7) had no knowledge about our visualization system until the case study sessions.

In general, all experts confirmed the importance of the three focused topics and appreciated our findings. E_2, E_4, and E_5 enjoyed the system's interactivity, especially the linked visualizations, which helped them connect the dots for comprehensive interpretations of important heads. Their existing visualization tools with piece-by-piece analysis fall short of such coordinated explorations. Using our system, E_1 and E_3 obtained an overview of ViTs' head attention patterns for the first time. Both experts found our findings intriguing and had thorough discussions on the analogy between CNNs and ViTs. For example, patches in Fig. 12(G) always attend to their right patch. The 3×3 CNN filter also aggregates the right pixel's value to the current pixel. So, the ViT head and CNN filter share equivalent functions. Similar equivalence analysis can also be extended to other heads/filters. E_7 pointed out similar learning trends across layers of CNNs and ViTs, i.e., lower-layer heads/filters focus on local features, whereas higher layers aggregate the output of lower layers to extract object-level information. E_7 also found the interactions between the `CLS` and patches very insightful. In Fig. 14(A), all patches attend to the patch below them and the last row attends to the `CLS`. This is similar to propagating the information top-down, and the final accumulated results are passed to `CLS` for classification. E_6 was interested in the content-agnostic/relevant attentions and believed they could be adopted for anomaly detection.

The experts also pointed out some insufficiency of the current system. First, E_7 initially thought only the three heads in Fig. 1(B1) were important to the prediction of image 712, which was misleading as we only pruned one head at a time and did not consider the dependency between heads. Second, both E_1 and E_7 worked on token pruning of transformers (instead of head pruning). They liked our image masks in disclosing the tokens' semantics but also wanted to see similar saliency maps highlighting the importance of individual patches through ablations. These comments provide promising future directions for us to explore.

VII. Discussion, Limitations, and Future Work

Despite many visual interpretation works for DL, *the unique values of our work* come from the following perspectives. First, our work presents a comprehensive interpretation of ViTs and discloses insightful findings. For example, heads with strong self-attentions are dominantly important. Lower- and higher-layer heads show different local/global attention strengths. Also, we summarize all possible attention patterns between patches. These insights open the hood of ViTs and deepen model designers' understanding. Second, our interpretation triggers model improvement ideas, e.g., pruning heads with repeating patterns. Thus, improving ViTs with our derived insights would be a direct follow-up work. Lastly, although we focus only on the classification task, we believe our interpretations are transferable

to ViT-based detection/generation tasks [31], as those tasks also significantly rely on the multi-head self-attentions of ViTs.

Head-Centric versus Image-Centric. We want to emphasize that all our analyses are *head-centric*, and each head's behavior is analyzed in one and across all images. Specifically, for *head importance*, we provide each head's *local* importance on one image and *global* importance over all images (Section V-B4). For *head attention strength*, we present a head's attention strengths in one image (Fig. 1(C2)) and its strength distribution over all images (Fig. 1(C3)). For *head attention pattern*, the two-axes/heatmap (Fig. 1(D3) and (D4)) shows the attention pattern of a head from one image, while the scatterplot in Fig. 1(D1) lays out the head's attention pattern over all images. From a different perspective, we believe *image-centric* analysis would also lead to insightful findings, e.g., checking if the heads show similar patterns for images of the same class. We plan to explore this direction in the future.

Performance. To guarantee the exploration interactivity, we have pre-computed some of the visualization data. For example, the head importance metrics are computed offline as they can take hours. The partial pruning in Fig. 1(B2) is performed online and each computation takes about 0.6 seconds on an Nvidia Titan RTX GPU. The head attention strengths and the tSNE layout for head attention patterns are both computed offline as they only need to be computed once and directly plugged into our system. In terms of storage, the raw attention weights consume the most space, ranging from 11 GB to 178 GB, depending on the number of heads in the studied ViT. Other data (e.g., images, probabilities, tSNE results) take about 300 MB in total.

Limitations and Future Work. Our head importance analysis relies on leave-one-out ablations, which do not consider the interaction between heads. In some cases, one head could be important only if another head is pruned. The analysis can be further extended to higher-order interactions, which is our planned future work. Second, our current analysis focuses on the attentions between two consecutive attention layers only. In the future, we would like to explore attention aggregation methods, e.g., [25], to interpret heads' impact across multiple layers. Lastly, we plan to investigate if the head importance, head attention strengths, and head attention patterns show any class-specific or dataset-specific trends. This will help to diagnose class-related performance issues and validate our findings in more datasets.

VIII. Conclusion

In this paper, we introduce a visual analytics solution to interpret ViTs. Our interpretation was carried out from three perspectives. First, we answer what heads are more important by introducing multiple head-importance metrics. Second, we explain why a head is important by disclosing its attention strength distribution across image patches in the 2D spatial context. Third, we adopt an unsupervised learning method to exhaustively summarize the possible attention patterns. Through concrete case studies conducted together with multiple experienced deep learning experts, we verify the efficacy of our visual interpretation solution.

References

[1] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[3] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[4] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[5] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Representations*, 2021.

[6] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14014–14024.

[7] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-attention attribution: Interpreting information interactions inside transformer," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 12 963–12 971.

[8] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky, "Revealing the dark secrets of BERT," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4365–4374.

[9] C. Park et al., "SANVis: Visual analytics for understanding self-attention networks," in *Proc. IEEE Visualization Conf.*, 2019, pp. 146–150.

[10] J. F. DeRose, J. Wang, and M. Berger, "Attention flows: Analyzing and comparing attention mechanisms in language models," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 1160–1170, Feb. 2021.

[11] R. Li, W. Xiao, L. Wang, H. Jang, and G. Carenini, "T3-Vis: Visual analytic for training and fine-tuning transformers in NLP," in *Proc. Conf. Empirical Methods Natural Lang. Process.: Syst. Demonstrations*, 2021, pp. 220–230.

[12] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 91–100, Jan. 2017.

[13] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 667–676, Jan. 2018.

[14] J. Wang, L. Gou, H. Yang, and H.-W. Shen, "GANViz: A visual analytics approach to understand the adversarial game," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 6, pp. 1905–1917, Jun. 2018.

[15] J. Wang, L. Gou, H.-W. Shen, and H. Yang, "DQNViz: A visual analytics approach to understand deep Q-networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 288–298, Jan. 2019.

[16] Z. Jin, Y. Wang, Q. Wang, Y. Ming, T. Ma, and H. Qu, "GNNLens: A visual analytics approach for prediction error diagnosis of graph neural networks," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2022.3148107.

[17] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 8, pp. 2674–2693, Aug. 2019.

[18] J. Choo and S. Liu, "Visual analytics for explainable deep learning," *IEEE Comput. Graph. Appl.*, vol. 38, no. 4, pp. 84–92, Jul./Aug. 2018.

[19] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics: Syst. Demonstrations*, 2019, pp. 37–42.

[20] T. Jaunet, C. Kervadec, R. Vuillemot, G. Antipov, M. Baccouche, and C. Wolf, "VisQA: X-raying vision and language reasoning in transformers," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 976–986, Jan. 2022.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[22] Z. Dong, T. Wu, S. Song, and M. Zhang, "Interactive attention model explorer for natural language processing tasks with unbalanced data sizes," in *Proc. IEEE Pacific Visualization Symp.*, 2020, pp. 46–50.

[23] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush, "Seq2seq-vis: A visual debugging tool for sequence-to-sequence models," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 353–363, Jan. 2019.

[24] E. Aflalo et al., "VL-InterpreT: An interactive visualization tool for interpreting vision-language transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 21 406–21 415.

[25] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4190–4197.

[26] S. Jin et al., "A visual analytics system for improving attention-based traffic forecasting models," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 1102–1112, Jan. 2023.
[27] J. Cao, Z. Gan, Y. Cheng, L. Yu, Y.-C. Chen, and J. Liu, "Behind the scene: Revealing the secrets of pre-trained vision-and-language models," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 565–580.
[28] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 12 116–12 128.
[29] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass, "Identifying and controlling important neurons in neural machine translation," in *Proc. Int. Conf. Learn. Representations*, 2019.
[30] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[31] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, 2021, Art. no. 200.

Yiran Li received the BS degree in mathematical sciences from Zhejiang University, in 2018. She is currently working toward the PhD degree in computer science with the University of California, Davis. Her research interests include visual analytics and interpretable machine learning.

Junpeng Wang received the BE degree in software engineering from Nankai University, the MS degree in computer science from Virginia Tech, and the PhD degree in computer science from the Ohio State University. He is a research scientist with Visa Research. His research interests include broadly in visualization, visual analytics, and explainable AI.

Xin Dai received the BE and MS degrees in computer science from Beijing Jiaotong University, in 2012 and 2016, respectively, and the PhD degree in computer science from Worcester Polytechnic Institute, in 2022. He is a staff research scientist with Visa Research. His research interests include data mining and machine learning.

Liang Wang received the BS degree in electrical engineering & automation and the MS degree in systems engineering, both from Tianjin University, and the PhD (highest honors) degree in computer science from Faculté Polytechnique de Mons, Mons, Belgium. He is a principal research scientist with Visa Research. His research interests include data mining, machine learning, and fraud analytics. Prior to joining Visa, he has worked with Yahoo!, eBay/PayPal, and FICO for bankcard fraud detection. He is the inventor of more than 20 patents and has published more than 30 papers in international journals and conferences.

Chin-Chia Michael Yeh received the PhD degree in computer Science from the University of California, Riverside. He is a staff research scientist with Visa Research. His PhD thesis "Toward a Near Universal Time Series Data Mining Tool: Introducing the Matrix Profile," received Doctoral Dissertation Award Honorable Mention at KDD 2019. He has published papers in top venues, including KDD, VLDB, ICDM and others. His research interests include data mining, machine learning, and time series analysis.

Yan Zheng received the PhD degree in computer science from the University of Utah, in 2017. She is currently a senior staff research scientist with Visa Research. She has published papers in top venues, including KDD, SIGMOD, ICDM and others. Her research interests include data mining, machine learning, and representation learning.

Wei Zhang received the bachelor's and master's degrees from the Department of Computer Science, Tsinghua University. He is a principal research scientist and research manager with Visa Research and interested in Big Data modeling and advanced machine learning technologies for payment industry. Prior to joining Visa Research, he worked as a research scientist with Facebook, R&D manager in Nuance Communications and also worked with IBM Research more than 10 years.

Kwan-Liu Ma (Fellow, IEEE) is a distinguished professor of computer science with the University of California, Davis. His research interests include the intersection of data visualization, computer graphics, human-computer interaction, and high performance computing. For his significant research accomplishments, he received several recognitions, recipient of the IEEE VGTC Visualization Technical Achievement Award in 2013, and inducted to IEEE Visualization Academy in 2019.