# **Visual Attention Exploration in Vision-Based Mamba Models**

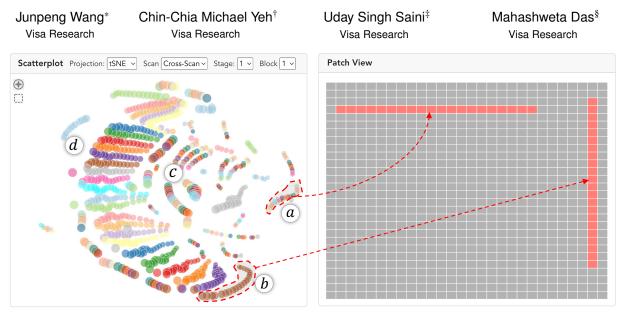


Figure 1: Our visual exploration tool contains two visualization components. The *Scatterplot* view on the left shows the dimensionality reduction results. The *Patch* view on the right shows the patch layout and highlights the patches of interest.

#### **A**BSTRACT

State space models (SSMs) have emerged as an efficient alternative to transformer-based models, offering linear complexity that scales better than transformers. One of the latest advances in SSMs, Mamba, introduces a selective scan mechanism that assigns trainable weights to input tokens, effectively mimicking the attention mechanism. Mamba has also been successfully extended to the vision domain by decomposing 2D images into smaller patches and arranging them as 1D sequences. However, it remains unclear how these patches interact with (or attend to) each other in relation to their original 2D spatial location. Additionally, the order used to arrange the patches into a sequence also significantly impacts their attention distribution. To better understand the attention between patches and explore the attention patterns, we introduce a visual analytics tool specifically designed for vision-based Mamba models. This tool enables a deeper understanding of how attention is distributed across patches in different Mamba blocks and how it evolves throughout a Mamba model. Using the tool, we also investigate the impact of different patch-ordering strategies on the learned attention, offering further insights into the model's behavior.

## 1 Introduction

State space models (SSMs) use state variables to mathematically describe the state of a dynamic system. They have a long history of modeling time series problems, where the state variables are time-dependent. Recent advances [4,5,12] have shown that SSMs achieve performance on par with state-of-the-art transformer models [3,13]. Additionally, their linear time complexity allows them to outperform transformers in latency-critical applications.

\*e-mail: junpenwa@visa.com †e-mail: miyeh@visa.com ‡e-mail: udasaini@visa.com §e-mail: mahdas@visa.com Mamba [4], also known as S6, is a cutting-edge SSM that enhances its predecessor S4 [5] with a selective scan mechanism. This mechanism enables the model to assign trainable weights to input tokens, allowing it to filter out less relevant information and emphasize more relevant details. These trainable weights are similar to the attention mechanism in transformers, which determines how much focus a token should place on other tokens. Since its release in late 2023, vision-based applications of Mamba have been rapidly developed. Similar to how transformers are adapted for vision tasks [3], vision-based Mamba models also decompose images into smaller patches and arrange them into sequences as inputs.

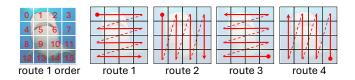


Figure 2: The four-way cross-scan in VMamba [10].

In vision transformers [3], all image patches are fed into the models simultaneously, with positional encoding used to differentiate patches at different positions. As a result, the order of patches has a relatively minor impact. However, in vision-based Mamba models, the patches are fed into the models sequentially one after another (similar to RNNs). This means that the order of the patches is crucial, as each patch can only collect information from its preceding ones. To ensure that each patch has access to all other patches, both forward and backward scans are often performed. For example, vision-Mamba (Vim) [17] uses routes 1 and 3 in Fig. 2 to allow each patch to "see" both its preceding and succeeding patches. VMamba [10] uses all four routes in Fig. 2 to preserve spatial locality between patches within the same row and column.

A vision-based Mamba model consists of a hierarchy of Mamba blocks, each of which learns the attention between image patches. However, it remains unclear (1) if the attention pattern is fixed within a block, (2) how attention patterns evolve across blocks, and (3) how

patch-ordering strategies impact the attention pattern. This paper seeks to address these questions using a visual analytics approach. Specifically, we extract the attention learned at each stage of the vision-based Mamba model and apply dimensionality reduction techniques to reveal attention patterns across stages. We also profile the attention patterns at each patch to disclose how attention is spatially distributed relative to the patch's position. Lastly, we introduce different patch-ordering strategies and compare their resulting attention patterns. In summary, our contributions are twofold:

- We design and develop a visual analytics tool to explore and summarize attention patterns in vision-based Mamba models.
- 2. We introduce multiple patch-ordering strategies and investigate their impact on patch attention in vision-based Mamba models.

#### 2 BACKGROUND AND RELATED WORK

Our work focuses on interpreting the attention mechanisms in visionbased Mamba models. Here, we briefly review the history of SSMs and summarize early work on attention interpretation.

#### 2.1 State Space Models (SSMs)

State space models (SSMs) are sequence modeling techniques that date back to the 1960s. They gained significant attention after Gu et al. [5] integrated the HiPPO matrix into them. The resulting model, known as S4, demonstrated substantial improvements in efficiency for modeling long sequences. However, the performance of S4 is not comparable to that of transformers, as it cannot effectively distinguish different input tokens. To overcome this, Gu and Dao [4] introduced the selective scan mechanism into S4, giving rise to S6, also known as Mamba. The selective scan allows Mamba to assign different weights to input tokens, enabling the model to learn the relative importance of each token. This allows Mamba to focus more on important tokens and selectively ignore less relevant ones. The learned weights are similar to the attention weights in transformers. Mamba demonstrates performance on par with transformers across a wide range of applications. More importantly, its linear complexity makes it a superior choice for many latency-critical applications, where the quadratic complexity of transformers would be prohibitive.

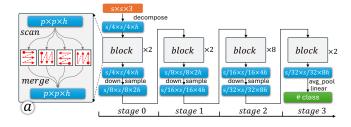


Figure 3: The architecture of VMamba for image classification [10].

In the vision domain, vision-Mamba (Vim) [17] and VMamba [10] are two seminal works that extend Mamba from NLP to vision tasks. Fig. 3 illustrates the architecture of VMamba for image classification. Given an RGB input image of size  $s \times s$ , the model first decomposes it into  $p \times p$  smaller patches. Each patch is considered as a token, and all patches are arranged into a sequence of length  $p \times p$ . This sequence is then treated similarly to textual data and fed into Mamba for learning. Specifically, the VMamba block in Fig. 3a employs four different scan orders (Fig. 2) to process the patches: (1) left $\rightarrow$ right then top $\rightarrow$ down, (2) top $\rightarrow$ down then left $\rightarrow$ right, (3) right $\rightarrow$ left then bottom $\rightarrow$ up, and (4) bottom $\rightarrow$ up then right $\rightarrow$ left. Each scan order introduces a unique dependency between patches, and consequently, the learned attention between patches also varies. The merge operation aggregates the attention learned for each patch across the four different scan orders. The output of this process

is a  $p^2 \times p^2$  attention matrix, where each element at position (i, j) represents the attention strength between patch i and patch j. Stacking multiple scan-merge blocks forms a stage, and Fig. 3 shows four such stages, each containing 2, 2, 8, and 2 blocks, respectively. Between stages, the latent representations are downsampled to distill essential features. Finally, a fully-connected layer transforms the latent representation into a vector, the length of which corresponds to the number of possible classes. Each element in this vector determines the probability for the corresponding class. Vim [17] follows a similar process, but uses only a two-way scan (routes 1 and 3 in Fig. 2). Therefore, we focus on VMamba in this work.

## 2.2 Attention Analysis and Visualization

Interpreting and diagnosing machine learning models is an important topic in the visualization literature [6,9,15], and there are multiple works focusing on interpreting the attention mechanisms of these models. For example, Abnar and Zuidema [1], Park et al. [11], and Li et al. [7] all used heatmaps to externalize the attention strengths in transformer-based NLP models. Vig [14] employed multiple parallel coordinate plots to visualize the attention patterns in BERT and GPT-2 models. DeRose et al. [2] introduced a radial layout for visualizing attention in BERT, layer by layer, which also facilitates comparisons between the attentions of two models. For vision transformers, Li et al. [8] used dimensionality reduction and scatterplots to summarize attention patterns within self-attention heads and across attention layers. Yeh et al. [16] employed a similar approach to visualize the joint q/k embedding space, providing a global view of attention patterns across a transformer model.

To the best of our knowledge, no studies have yet focused on analyzing the attention patterns in Mamba models. In particular, we are interested in whether regular attention patterns emerge within a Mamba block and whether hierarchical attention patterns develop across different stages of a Mamba model. To explore these questions, we have developed a visual analytics tool specifically designed to analyze attention patterns in vision-based Mamba models.

### 3 PROBLEM AND METHODOLOGY

In this study, we aim to investigate the attention patterns in Mamba, focusing on two key aspects based on insights from existing literature and the interests of domain experts working with Mamba:

- Inter-Block Attention Pattern: Are the attention patterns consistent across different Mamba blocks within the same stage?
- Intra-Block Attention Pattern: How is the attention spatially distributed across image patches within a single Mamba block?

We provide solutions to answer these two questions in Sec. 3.1 and Sec. 3.2, respectively. For our study, we focus on the VMamba model [10], whose architecture is illustrated in Fig. 3. The model is trained for 400 epochs on the ImageNet dataset (1,281,167 training images). After training, we use 1000 test images to investigate the attention patterns. The input images are of size  $224 \times 224 \times 3$  (i.e., s=224 in Fig. 3) and the number of patches at the four stages is  $56 \times 56$ ,  $28 \times 28$ ,  $14 \times 14$ , and  $7 \times 7$ , respectively.

## 3.1 Inter-Block Attention Pattern

The attention matrices for different Mamba blocks at the same stage have the same size. For example, in Fig. 3, the attention matrices from the two blocks at stage 0 are of the same size, which is  $56^2 \times 56^2$  when s=224. This allows us to compare the matrices and assess how the attention patterns differ across blocks at the same stage. However, attention matrices from blocks across different stages have varying sizes, and thus, these blocks are not directly comparable.

When an image is fed through a Mamba block, it produces an attention matrix of size  $p^2 \times p^2$ , where the element at position (i, j) represents the attention strength from patch i to patch j. For example,

Algorithm 1 Generating cluster pattern across blocks of a stage. Require: n, m> number of images and blocks Require: stage be the focused stage Require: image b the array of images **Require:** VMamba, DR  $\triangleright$  the VMamba model and DR algorithm 1: attentions = []2: **for**  $i \leftarrow 1$  to n **do** 3: **for**  $j \leftarrow 1$  to m **do** 4: attention = VMamba(image[i], stage, j)5: attentions.append(attention) 6: end for 7: end for  $\triangleright points \in \mathbb{R}^{m \times n \times 2}$ 8: points = DR(attentions)

at stage 0 in Fig. 3, the attention matrix will be of shape  $56^2 \times 56^2$ . If the focused stage contains m blocks, we will obtain m such  $p^2 \times p^2$  matrices. By feeding all n=1000 test images through the VMamba model and collecting their attention matrices from the m blocks at the focused stage, we compile the attention data into a large matrix of shape  $(m \times n) \times (p^2 \times p^2)$ . This process is outlined in lines 1 to 7 of the pseudo-code in Algorithm 1.

We then apply dimensionality reduction (DR) techniques to map the  $m \times n$  points, originally in the  $p^2 \times p^2$  dimensional space, to a 2D space for visualization as a scatterplot. If the n points from different blocks form isolated clusters, this indicates that the attention patterns between the blocks are significantly different. The details of the visualization and interactions are described later in Sec. 4.1.

#### 3.2 Intra-Block Attention Pattern

According to the formulation in Sec. 3.1, the attention matrix for a single block and a single image has the shape  $p^2 \times p^2$  (line 4 of Algorithm 1). The *i*-th row of this matrix represents the attention strength from patch *i* to all  $p^2$  patches. To identify patches with similar attention patterns, we can apply dimensionality reduction (DR) techniques to reduce the  $p^2 \times p^2$  matrix to a  $p^2 \times 2$  matrix, which can then be analyzed through a scatterplot visualization to assess the similarity between patches.

However, the approach described above can be significantly influenced by the image content, making it difficult to extract common, content-agnostic patterns. To enhance the identification of common attention patterns, we generate the  $p^2 \times p^2$  attention matrix for all n images and aggregate the resulting n matrices into a single  $p^2 \times p^2$  matrix for pattern augmentation. The details of this process are outlined in Algorithm 2 (lines 1 to 6).

We employ a scatterplot to visualize the  $p^2$  2D points. The color and size of each point correspond to the column and row of the respective patch. By examining the clustering patterns of these points, we can identify patches with similar attention patterns and relate these similarities to their spatial position in the image space (detailed in Sec. 4.1).

Algorithm 2 Generating cluster pattern within a block.

```
Require: n
                                                Require: stage, block
                                      be the focused stage and block
Require: image
                                               b the array of images
Require: VMamba, DR 
ightharpoonup the VMamba model and DR algorithm
 1: attentions = []
 2: for i \leftarrow 1 to n do
        attention = VMamba(image[i], stage, block)
 3.
        attentions.append(attention)
 4:
 5: end for
 6: avg\_attn = np.mean(np.array(attentions), axis = 0)
                                                   \triangleright points \in \mathbb{R}^{p^2 \times 2}
 7: points = DR(avg\_attn)
```

#### 4 VISUAL ANALYTICS SYSTEM

We have developed a visual analytics tool to effectively visualize the outputs of Algorithm 1 and Algorithm 2. The tool features two distinct visualization views: the *Scatterplot* view (Fig. 1, left) and the *Patch* view (Fig. 1, right).

## 4.1 The Scatterplot View

The *Scatterplot* view provides two visualization modes for exploring the inter-block and intra-block attention patterns.

Mode 1: Users can select a stage and a block from the header of the *Scatterplot* view (Fig. 1, left). If the block ID is not specified, the *Scatterplot* will display  $m \times n$  points based on the output of Algorithm 1, where m represents the number of blocks in the selected stage, and  $n{=}1000$  is the number of images. Each point corresponds to a  $p^2 \times p^2$  attention matrix for a given image from a specific block. For example, in the VMamba architecture depicted in Fig. 3, there are 4 stages, with 2, 2, 8, and 2 blocks per stage, respectively. The dimensionality reduction results for the four stages are shown in Fig. 4, where each figure represents the output of a single stage. The color in each figure corresponds to the block ID, and distinct clusters of points are clearly visible, with different colors separating the blocks. This indicates that blocks within the same stage exhibit noticeably different attention patterns.

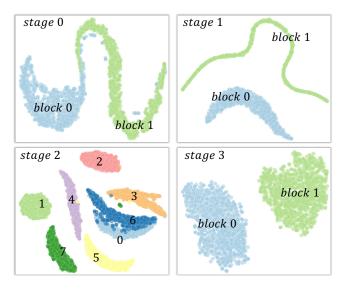


Figure 4: Attention pattern similarity between blocks from the four stages of the VMamba model in Fig. 3.

Mode 2: When a block ID is specified, the *Scatterplot* will show  $p^2$  points based on the output of Algorithm 2. In this case, each point represents the averaged attention across n=1000 test images for a specific patch position. The color and size of each point correspond to the column and row of the corresponding patch, respectively. As illustrated by the clustering pattern in Fig. 1, patches from the same row (points in Fig. 1a with the same size) or the same column (points in Fig. 1b with the same color) exhibit similar attention patterns. Figs. 1c and 1d reveal further clusters that share similar attention patterns to those in Figs. 1a and 1b, respectively.

For both modes, we provide three popular dimensionality reduction techniques: PCA, tSNE, and UMAP. Each technique has its unique strengths and limitations, so offering multiple options allows users to explore different perspectives and uncover subtle clustering patterns. When processing very high-dimensional data using tSNE and UMAP, we first use PCA to project them to a relatively lower dimension (i.e., 100D), then apply tSNE and UMAP for efficiency.

The *Scatterplot* view also supports zooming, enabling users to examine cluster details at various levels of granularity. In Mode 2, users can perform lasso selection to highlight a group of points. The corresponding patches for the selected points will then be visualized in the *Patch* view (described in the following section).

#### 4.2 The Patch View

The *Patch* view also offers two visualization modes, allowing users to explore patch-level details from the selected stage.

Mode 1: In this mode, the patches from the selected stage are displayed as gray squares. For example, in Fig. 1 (right), the 28×28 patches from stage 1 are visualized as 28×28 squares. Meanwhile, the selected patches from the *Scatterplot* view (Fig. 1, left) are highlighted as red squares in the *Patch* view, visually indicating their spatial location within the image. This highlighting directly corresponds to the cluster patterns observed in the *Scatterplot* view (Fig. 1, left), which helps reveal the spatial relationships between patches that exhibit similar attention patterns. This coordinated visualization provides a clear understanding of how attention patterns are distributed across different regions of the image.

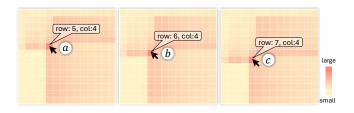


Figure 5: Patches in the same column exhibit similar attention.

Mode 2: When a square/patch is clicked in the *Patch* view, all squares/patches will be colored according to their attention strength to the clicked one. Attention strengths from small to large are mapped to colors from light-yellow to dark-red. For example, in Fig. 5a, the clicked patch is located at row 5, column 4. This patch shows strong attention to: (1) itself, (2) the patches in the same row to its left, and (3) the patches to its right. From the *Scatterplot* view, we noticed that patches in the same column as the selected patch have similar attention patterns. Based on this, we select two additional patches for further exploration. As shown in Figs. 5b and 5c, these two patches, which are in the same column as the one in Fig. 5a, exhibit very similar attention patterns.

#### 5 FINDINGS AND PATTERN SUMMARY

Using the developed visual analytics tool, we conducted a detailed exploration of the four stages and their respective blocks in the VMamba model (Fig. 3). This section summarizes the key findings from our analysis.

Finding 1: Blocks within the same stage exhibit significantly different attention patterns. This finding is clearly illustrated in Fig. 4. To examine the differences further, we focus on the two blocks at stage 0 for a detailed exploration. As shown in Figs. 6-a1 (stage 0, block 0) and b1 (stage 0, block 1), the color and size of the points (representing individual patches) change progressively along both the row and column directions. This indicates a gradual transition of the attention pattern for patches along rows and columns. When individual patches are clicked in the Patch view, their attention patterns are displayed on the right. It is evident that the same patch from the two blocks exhibits very different attention patterns. In block 0 (Fig. 6-a2), the selected patch strongly attends to (1) itself, (2) patches in the first row with a larger column ID, (3) patches in the first column with a larger row ID, and (4) patches that have both larger row and column IDs than the selected patch. In contrast, the

patch in block 1 (Fig. 6-b2) pays noticeably weaker attention to itself and patches along similar rows and columns, but stronger attention to patches in other areas of the image. We checked multiple patches from these two blocks, and the observation is consistent across them.

Finding 2: Some blocks exhibit complementary attention patterns. For instance, the selected patch at stage 2, block 1 (Fig. 6-e2) shows stronger attention to itself as well as to patches in the same row and column. In contrast, the same patch at stage 2, block 3 (Fig. 6-f2) and block 5 (Fig. 6-g2) exhibits weaker attention to the patch itself and to patches along the same row and column. This complementary attention behavior across blocks enables VMamba to selectively focus on relevant patches, contributing to its flexibility in attending to different regions of the input images.

Finding 3: There is a hierarchy regarding the attention pattern learned from early to later stages. At early stages, patches that are spatially closer tend to exhibit similar attention patterns, while at later stages, the attention is more influenced by the image content. At stage 0 (Fig. 6, a1-b1), attention patterns show smooth and progressive changes for patches in the same row and column, indicating a strong spatial correlation. At stage 1 (Fig. 6, c1-d1), clear clusters form for patches in the same row and column—represented by points of the same color but varying sizes or the same size but different colors. This pattern persists at stage 2 (Fig. 6, e1-h1), though clustering is less pronounced for some blocks at this stage. By stage 3, the clustering structure becomes less obvious in the Scatterplot view (Fig. 6, i1-j1), suggesting that the attention patterns have become more diverse and content-dependent. This phenomenon aligns with the behavior observed in CNNs and vision transformers, where lower layers tend to focus on local, content-agnostic features, while higher layers capture more complex, content-relevant patterns.

#### 6 IMPACT OF PATCH ORDER ON ATTENTION PATTERNS

One of our key findings is that patches within the same row or column often exhibit similar attention patterns. We hypothesize that this is influenced by the order in which patches are arranged into sequences. To test this hypothesis, we introduce three alternative patch orders, shown in Fig. 7, and examine how the attention patterns change when these new orders are applied.

Fig. 7a shows a patch order that scans patches along the diagonal. Fig. 7b employs the Morton order (z-order curve), a well-known space-filling curve that is particularly effective at preserving spatial locality. Fig. 7c arranges patches in a spiral layout, where the innermost patch retains the highest spatial locality. We modified the VMamba code to implement each of these patch orders and trained the model from scratch for 400 epochs. All three scanning methods achieved accuracy levels similar to the original VMamba model, i.e., achieving accuracy greater than or equal to 82.6% on ImageNet as reported in the original VMamba paper [10].

Figs. 8-a1, a2, and a3 show the results of exploring attention patterns using the diagonal order. In Fig. 8-a1, all patches are projected onto a continuous curve. By selecting points in a local region, we observe that the corresponding patches are adjacent along the diagonal, as shown in Fig. 8-a2. When clicking on any patch in the *Patch* view, we see that its attention behavior mirrors what was observed in Fig. 6 but along the diagonals instead of the rows or columns. For example, the attention pattern of a patch at stage 1, block 0 is displayed in Fig. 8-a3. This patch strongly attends to its preceding patches along the diagonal, which is very similar to what was observed in Fig. 6-c2. However, in Fig. 6-c2, the preceding patches are those based on the cross-scan order in Fig. 2.

Next, we conduct similar explorations with the VMamba model trained using the Morton order. As shown in Fig. 8-b1, patches at stage 1, block 0 are grouped into clusters. Selecting a cluster highlights the patches within a local region in Fig. 8-b2. These patches are actually in the same z-order curve block. The strong spatial locality among them results in similar attention patterns. Clicking on a

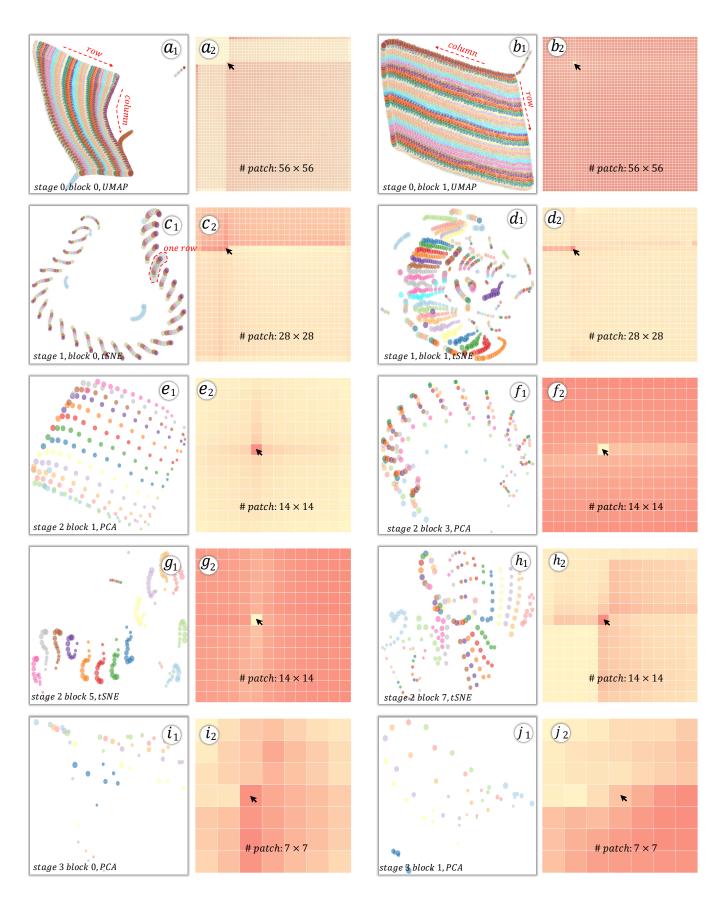


Figure 6: The attention pattern similarity between patches from blocks of different stages. (a1-j1) Two, two, four, and two blocks from stages 0, 1, 2, and 3 are shown, respectively. (a2-j2) Selecting a patch to inspect its attention pattern.

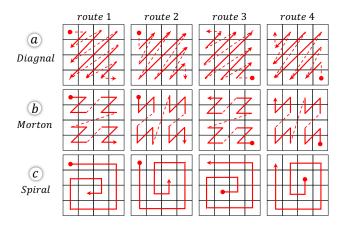


Figure 7: Arranging patches in VMamba following different orders.

patch reveals that it also strongly attends to its preceding patches, as shown in Fig. 8-b3. The results for the spiral patch order, shown in Figs. 8-c1, c2, and c3, are consistent with those observed from other patch orders. These findings confirm that stage 1, block 0 of the VMamba model exhibits a fixed attention pattern: any patch in this block consistently attends strongly to its preceding patches. However, the definition of "preceding patches" depends on the specific patch order used.

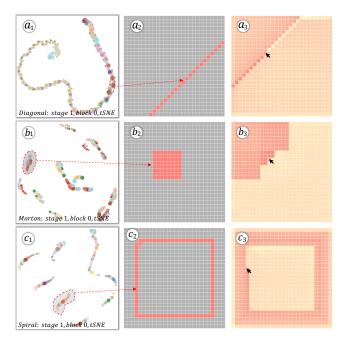


Figure 8: Exploring attention patterns when patches are arranged following the three new orders in Fig. 7: Diagonal, Morton, and Spiral.

#### 7 CONCLUSION AND FUTURE WORK

In this paper, we introduced a visual analytics tool to explore and compare attention patterns within and across VMamba blocks. Through our exploration, we discovered several key insights: (1) VMamba blocks within the same stage exhibit distinct attention patterns; (2) the order of patches significantly influences the resulting attention patterns; and (3) patches that are close in the input sequence generally exhibit similar attention patterns. Given the significant impact of patch arrangement on attention patterns, we

proposed multiple new patch orders that better preserve the patches' spatial locality. Using our tool, we further investigated VMamba models trained with these new orders, and found similar attention behaviors to those observed in the original patch order.

Looking ahead, we aim to extend our tool in several directions. First, we plan to explore more complex VMamba models with additional stages and blocks per stage. We hypothesize that the general trends observed in this study will hold as the architecture becomes more intricate. Second, our current work focuses on identifying content-agnostic attention patterns by averaging attention over n images. In the future, we intend to integrate additional views into the tool to present content-relevant attention patterns, which will be particularly useful for diagnosing specific images of interest.

#### REFERENCES

- S. Abnar and W. Zuidema. Quantifying attention flow in transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4190–4197, July 2020.
- [2] J. F. DeRose, J. Wang, and M. Berger. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Trans. Vis. Comput. Graphics*, 27(2):1160–1170, 2020.
- [3] 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. In 9th International Conference on Learning Representations (ICLR), 2021.
- [4] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [5] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.
- [6] 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, 2018.
- [7] R. Li, W. Xiao, L. Wang, H. Jang, and G. Carenini. T3-vis: visual analytic for training and fine-tuning transformers in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 220–230, 2021.
- [8] Y. Li, J. Wang, X. Dai, L. Wang, C.-C. M. Yeh, Y. Zheng, W. Zhang, and K.-L. Ma. How does attention work in vision transformers? a visual analytics attempt. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):2888–2900, 2023.
- [9] S. Liu, W. Yang, J. Wang, and J. Yuan. Visualization for Artificial Intelligence. Springer Cham, 2025. doi: 10.1007/978-3-031-75340-4
- [10] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu. VMamba: Visual state space model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [11] C. Park, I. Na, Y. Jo, S. Shin, J. Yoo, B. C. Kwon, J. Zhao, H. Noh, Y. Lee, and J. Choo. Sanvis: Visual analytics for understanding selfattention networks. In 2019 IEEE Visualization Conference (VIS), pp. 146–150. IEEE, 2019.
- [12] J. T. Smith, A. Warrington, and S. W. Linderman. Simplified state space layers for sequence modeling. arXiv preprint arXiv:2208.04933, 2022.
- [13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems, 2017.
- [14] J. Vig. A multiscale visualization of attention in the transformer model. arXiv preprint arXiv:1906.05714, 2019.
- [15] J. Wang, S. Liu, and W. Zhang. Visual analytics for machine learning: A data perspective survey. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7637–7656, 2024.
- [16] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas, and M. Wattenberg. Attentionviz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics*, 30(01):262–272, 2024.
- [17] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417, 2024.