Samba: A Unified Mamba-based Framework for General Salient Object Detection

Jiahao He[1] Keren Fu[1,3,*] Xiaohong Liu[2] Qijun Zhao[1,3]
[1]College of CS, Sichuan University [2]John Hopcroft Center, Shanghai Jiao Tong University
[3]National Key Lab of Fundamental Science on Synthetic Vision, Sichuan University

Abstract

*Existing salient object detection (SOD) models primarily resort to convolutional neural networks (CNNs) and Transformers. However, the limited receptive fields of CNNs and quadratic computational complexity of transformers both constrain the performance of current models on discovering attention-grabbing objects. The emerging state space model, namely Mamba, has demonstrated its potential to balance global receptive fields and computational complexity. Therefore, we propose a novel unified framework based on the pure Mamba architecture, dubbed saliency Mamba (**Samba**), to flexibly handle general SOD tasks, including RGB/RGB-D/RGB-T SOD, video SOD (VSOD), and RGB-D VSOD. Specifically, we rethink Mamba's scanning strategy from the perspective of SOD, and identify the importance of maintaining spatial continuity of salient patches within scanning sequences. Based on this, we propose a saliency-guided Mamba block (SGMB), incorporating a spatial neighboring scanning (SNS) algorithm to preserve spatial continuity of salient patches. Additionally, we propose a context-aware upsampling (CAU) method to promote hierarchical feature alignment and aggregations by modeling contextual dependencies. Experimental results show that our **Samba** outperforms existing methods across five SOD tasks on 21 datasets with lower computational cost, confirming the superiority of introducing Mamba to the SOD areas. Our code is available at https://github.com/Jia-hao999/Samba.*

1. Introduction

Salient object detection (SOD) is an essential vision task that aims to identify and segment the most visually prominent objects within a scene. This technique plays a crucial role in various applications such as object tracking [90], semantic segmentation [14], image enhancement [51], autofocus [28] and evaluation of large models [29].

Current state-of-the-art (SOTA) SOD methods are primarily dominated by convolutional neural networks (CNNs) and transformers, addressing various SOD tasks, including RGB/RGB-D/RGB-T SOD [4, 23, 40, 62, 63, 72], video SOD (VSOD) [21, 80, 85], and RGB-D VSOD [36, 39, 53]. While CNN-based backbones are known for their scalability and linear computational complexity, they suffer from limited receptive fields, making it challenging to capture global dependencies. In contrast, transformer-based backbones offer superior visual modeling by leveraging global receptive fields. However, their self-attention mechanism incurs quadratic complexity, raising efficiency concerns. Although efficient transformer architectures, such as Swin transformer [44] and MobileViT [50], have been proposed, they typically sacrifice some global modeling capability for efficiency, failing to achieve an optimal balance between the two.

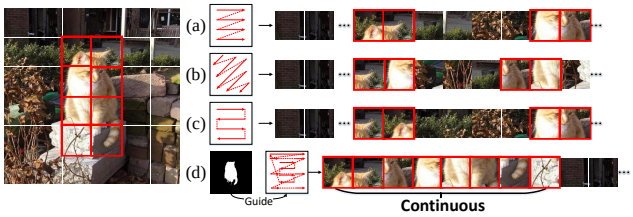

Figure 1. Comparison between existing scanning strategies and our scanning strategy. (a) Sequential scanning of patches in a "Z" pattern [43]. (b) Sequential scanning of patches in diagonal directions [60, 84]. (c) Sequential scanning of patches in an "S" pattern [76]. (d) Compared to (a/b/c), our spatial neighboring scanning (SNS) can preserve spatial continuity of salient patches.

Recently, Mamba [16], a novel state space model (SSM), has emerged as a highly promising backbone to balancing global receptive fields and computational efficiency. Mamba employs a selective scanning algorithm to model long-range dependencies while preserving linear complexity. Besides, with a specially designed hardware-aware algorithm, Mamba achieves efficient training on GPUs. Building on this foundation, visual Mamba backbones [43, 92], with task-specific models [6, 68, 78, 87, 91] based on them, have been rapidly developed. Given Mamba's success across various vision tasks and its absence in SOD areas, we seek to explore its potential for efficient global modeling in SOD tasks.

In this paper, we propose a novel unified model, saliency Mamba (*Samba*), to flexibly handle general SOD tasks. Owing to the strong performance of visual Mamba backbones, many task-specific models [49, 68, 78, 91] have leveraged

*Corresponding author: Keren Fu *(fkrsuper@scu.edu.cn)*.

them to extract global cues. Inspired by this, we adopt Vmamba [43] as our backbone, and attempt to design a Mamba-based decoder to produce elaborated results. In our approach, we refer to existing Mamba-based decoders [20, 49, 68, 74, 76, 84], and identify two crucial issues for designing Mamba-based SOD decoders:

Spatial continuity issue. In the workflow of visual Mamba models, 2D feature maps are first divided into patches and then scanned into 1D sequences before being fed to SSMs. Since SSMs are designed to provide continuous predictions for 1D causal sequences, the prediction of a current image patch heavily relies on preceding patches, especially the nearest ones. Therefore, continuous (or successive) salient patches within 1D sequences can help SSMs accurately locate complete salient regions, enhancing feature representation. However, previous scanning strategies [43, 60, 76, 84] neglect this issue and fail to maintain spatial continuity of salient patches (as shown in Fig. 1 (a), (b) and (c)), hindering SSMs to generate high-quality features.

Feature alignment issue. Existing Mamba-based decoders [49, 68, 91] typically employ nearest-neighbor interpolation to upsample low-resolution (i.e., high-level) features before incorporating them with high-resolution (i.e., low-level) features. However, this approach leads to two limitations: 1) it lacks learnability; 2) it neglects the contextual dependencies between hierarchical features. These issues result in misalignment during feature fusion, causing deviations in the final prediction. Although previous works [42, 64] have proposed learnable upsampling methods, they still fail to model the contextual dependencies between hierarchical features during the upsampling process.

To address the **first issue**, we propose a novel saliency-guided Mamba block (SGMB), which emphasizes spatial continuity of salient patches, and leverage SSM's global modeling capability to enhance feature representation. Specifically, we design a spatial neighboring scanning (SNS) algorithm to generate scanning paths, which are applied to flatten 2D feature maps into 1D sequences while preserving spatial continuity of salient patches (Fig. 1 (d)). These 1D sequences are then processed by SSMs to generate high-quality features. Notably, compared to commonly fixed scanning strategies (Fig. 1 (a/b/c), SNS can dynamically tune scanning directions to handle various scenarios, offering insights for future designs of scanning strategies. To tackle the **second issue**, we propose a context-aware upsampling (CAU) method, with a novel patch pairing and ordering scheme, to promote hierarchical feature alignment and aggregations during decoding. First, patches from shallow and deep features are paired as subsequences to model the contextual dependencies between hierarchical features. These paired subsequences are then concatenated and input to SSMs. By leveraging powerful causal prediction of SSMs, deep features can progressively learn data distributions of shallow features, and then are expanded to the same shapes as shallow features for fusion. To flexibly handle general SOD tasks, we also propose a multi-modal fusion Mamba (MFM) to explore the interaction and integration of multi-modal information.

In a nutshell, this paper provides four main contributions:

- To the best of our knowledge, we are the first to adapt state space models to SOD tasks, and propose a novel unified framework based on the pure Mamba architecture to flexibly handle general SOD tasks.
- We propose a saliency-guided Mamba block (SGMB), incorporating a spatial neighboring scanning (SNS) algorithm, to maintain spatial continuity of salient patches, thus enhancing feature representation.
- We propose a context-aware upsampling (CAU) method to promote hierarchical feature alignment and aggregations by modeling contextual dependencies.
- Our *Samba* achieves SOTA results across five SOD tasks on 21 datasets, validating its effectiveness as well as the potential of introducing Mamba to the SOD areas.

2. Related Work

2.1. Deep Learning based SOD

RGB SOD. Initially, SOD researches focus solely on the RGB modality, and design various methods, such as boundary enhancement [31, 83], feature refinement [73, 86] and attention mechanism [72, 86]. Recently, transformer-based methods [40, 48, 93] have become mainstream, offering superior performance due to their powerful global modeling. However, these methods fail to tackle some challenging scenes, such as complex and low-contrast backgrounds.

RGB-D and RGB-T SOD. To address the challenging scenes, some works introduce depth [3, 12, 23, 24, 32, 63, 88] or thermal images [2, 4, 25, 65, 81, 82] to assist saliency detection. For example, Fu et al. [12, 13] utilize a Siamese network to extract shared information from RGB and depth inputs for more accurate detection in complex scenes. Tu et al. [65] propose a novel dual-decoder architecture to integrate the multi-type interactions between RGB and thermal.

VSOD. In contrast to static images, dynamic video scenarios present considerable difficulties due to the diversity of motion patterns. Therefore, leveraging temporal contexts within video sequences is crucial for VSOD methods. Some studies [19, 21, 80, 85] use the interaction between video frames to extract motion cues. Other researches [26, 35, 41, 59] precompute optical flow maps between adjacent frames, and then extract complementary motion information from them.

RGB-D VSOD. The effectiveness of integrating depth into VSOD models has been demonstrated in [46], giving rise to a potential research direction: RGB-D VSOD. As acquiring RGB-D videos becomes easier, RGB-D video datasets have been introduced [36, 39, 53]. At the same time, RGB-D VSOD methods such as DCTNet+ [53] and ATFNet [39]

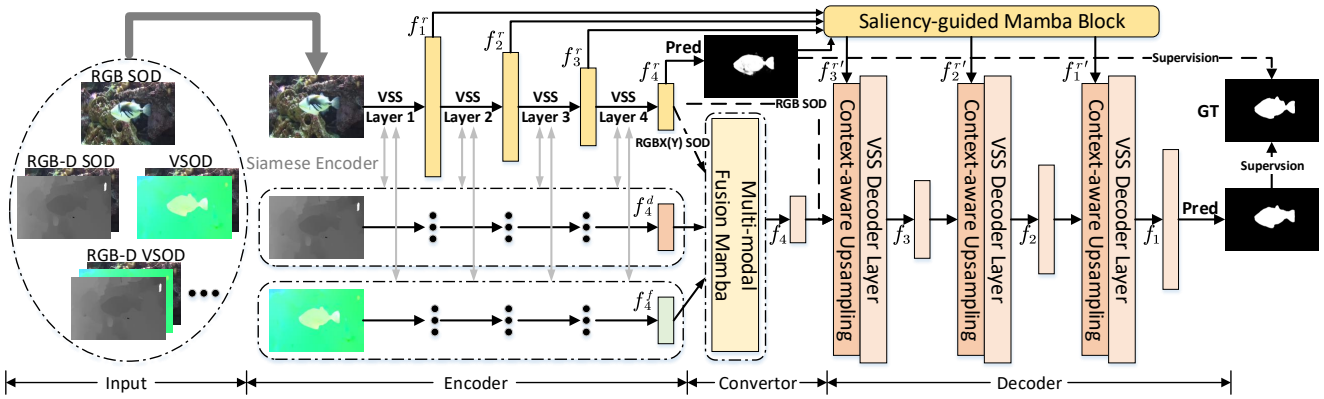

Figure 2. Overall architecture of the proposed *Samba* model for general SOD tasks.

have shown encouraging potential in detection performance.

2.2. Visual Mamba

Motivated by the success of Mamba in language modeling, Zhu et al. [92] transfer this success to vision, and design an efficient visual backbone, vision Mamba (Vim), which incorporates the bidirectional SSMs for global context modeling. Liu et al. [43] design visual state space blocks, and based on them, a novel visual Mamba backbone (Vmamba) is developed and demonstrates promising performance across a range of vision tasks, including semantic segmentation [68, 91] and object detection [6, 87]. Due to the impact of the SSM's selective scanning direction on effective receptive fields, Zhao et al. [84] expand on VMamba's four-directional scanning by adding four additional diagonal directions, i.e., Fig. 1 (b), to extract large spatial features from multiple directions. Yang et al. [76] propose PlainMamba, with a continuous 2D scanning approach, i.e., Fig. 1 (c), ensuring that the scanning sequences are spatially continuous.

3. Methodology

3.1. Preliminaries

SSMs [17, 18, 61] are sequence-to-sequence models designed to capture long-range dependencies using linear time-invariant (LTI) systems. These systems map an input sequence $x(t) \in \mathbb{R}$ to an output sequence $y(t) \in \mathbb{R}$ via a latent state $h(t) \in \mathbb{R}^N$, represented by the following ordinary differential equations (ODEs):

$$y(t) = \mathbf{C}h(t) + \mathbf{D}x(t), h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times N}$, and $\mathbf{D} \in \mathbb{R}^1$ are system parameters. $h'(t)$ is the time derivative of $h(t)$.

The discretization of this continuous-time system is essential for integrating SSMs into deep learning frameworks. A common method for discretization is zero-order hold (ZOH) [17], which can be formulated as:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}), \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B}, \quad (2)$$

where $\Delta \in \mathbb{R}^D$ is a predefined timescale parameter.

This process allows the discrete-time SSM to be expressed in a recurrent form, mapping the input sequence $\{x_1, x_2, ..., x_k\}$ to the output sequence $\{y_1, y_2, ..., y_k\}$:

$$y_k = \mathbf{C}h_k + \mathbf{D}x_k, h_k = \overline{\mathbf{A}}h_{k-1} + \overline{\mathbf{B}}x_k. \quad (3)$$

SSMs efficiently handle long-range dependencies with linear complexity, but their time-invariance limits capturing dynamic context. To address this, the recent advancement, Mamba [16], introduce an input-dependent selection mechanism (S6) that relaxes the time-invariance constraint. Mamba allows the system parameters, specifically $\mathbf{B}$, $\mathbf{C}$, Δ, to vary based on the input, thereby enhancing the model's flexibility in capturing complex interactions within long sequences. Mamba also employs a new parallel scanning algorithm, enabling efficient training and inference on GPUs.

3.2. Big Picture

In this section, we present an overview of the proposed *Samba* model for general SOD tasks, as shown in Fig. 2. The "Input" encompasses various SOD tasks that *Samba* can adapt to, with the "..." indicating other potential tasks not explicitly listed, such as RGB-T SOD. The "Encoder" employs a Siamese backbone that contains four visual state space (VSS) layers [43], to extract multi-level features from the inputs. The "Convertor" integrates information from different modalities through a multi-modal fusion Mamba (MFM). For the "Decoder", we mainly propose a novel saliency-guided Mamba block (SGMB) and a context-aware upsampling (CAU) method. SGMB employs a novel scanning strategy to maintain spatial continuity of salient patches, thereby enhancing feature representation. CAU is designed to facilitate the alignment and aggregations of hierarchical features by modeling contextual dependencies. In the subsequent sections, we will provide detailed descriptions of the "Encoder", "Convertor", and "Decoder".

3.3. Encoder

To address general SOD tasks, we implement a Siamese encoder based on VSS layers. The encoder begins by partitioning the input images into patches. Then four VSS layers,

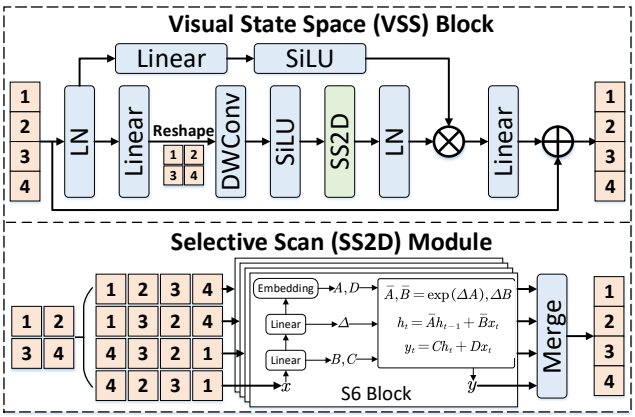

Figure 3. Diagram of the visual state space (VSS) block and selective scan (SS2D) module.

each containing multiple VSS blocks and a downsampling operation, are cascaded to extract multi-level features f_i^m, where $m \in [r, d, f, t]$ represent RGB, depth, optical flow and thermal modalities, respectively, and $i \in [1, 2, 3, 4]$ denotes the layer index. Fig. 3 provides a detailed illustration of the VSS block and its core selective scan (SS2D) module.

VSS. The input first undergoes a layer normalization (LN), after which it is split into two information flows. The first flow is processed by a sequence of operations: a linear projection (Linear), a reshape operation (Reshape), a depth-wise convolution (DWConv) and a SiLU activation function [7]. Next, an SS2D module is applied to model global dependencies, followed by another LN layer. The second flow, by contrast, only passes through a Linear and a SiLU activation function. After that, the two flows are multiplied and processed by a Linear layer for an output. Finally, the original input is added to the output via a residual connection.

SS2D. The input 2D feature is first flattened into four 1D sequences by scanning along four distinct directions. Then the four sequences are processed by S6 blocks [16] to capture long-range dependencies. Lastly, the sequences are reordered into the same direction, and then summed to merge information.

3.4. Convertor

To facilitate flexible extension from single-modal SOD (RGB SOD) to dual-modal (RGB-D/T SOD, VSOD) and tri-modal SOD (RGB-D VSOD), we design a multi-modal fusion Mamba (MFM) as the convertor and insert it between the encoder and decoder. When handling the RGB SOD task, the convertor remains empty, and f_4^r is directly fed to the decoder. Within the dual-modal convertor, $f_4^r \in \mathbb{R}^{H_4 \times W_4 \times C_4}$ and $f_4^x \in \mathbb{R}^{H_4 \times W_4 \times C_4}$, where $x \in [d, f, t]$, are first processed by a Linear and a DWConv, respectively. The outputs are then flattened into $\mathbb{R}^{L \times C_4}$, where $L = H_4 \times W_4$, and concatenated along the L dimension. To explore the interaction of multi-modal information, we utilize an S6 block to process the concatenated sequence. Finally, the sequence is split into two outputs, which are summed and processed by a Linear projection. This process can be formulated as:

$$\begin{aligned} \bar{f}_4^r &= DWConv\left(Linear\left(f_4^r\right)\right), \\ \bar{f}_4^x &= DWConv\left(Linear\left(f_4^x\right)\right), \\ \tilde{f}_4^r, \tilde{f}_4^x &= Split\left(S6\left(Cat\left(\bar{f}_4^r, \bar{f}_4^x\right)\right)\right), \\ f_4 &= Linear\left(\tilde{f}_4^r + \tilde{f}_4^x\right). \end{aligned} \tag{4}$$

Building upon the dual-modal convertor, we can seamlessly extend it to a tri-modal convertor.

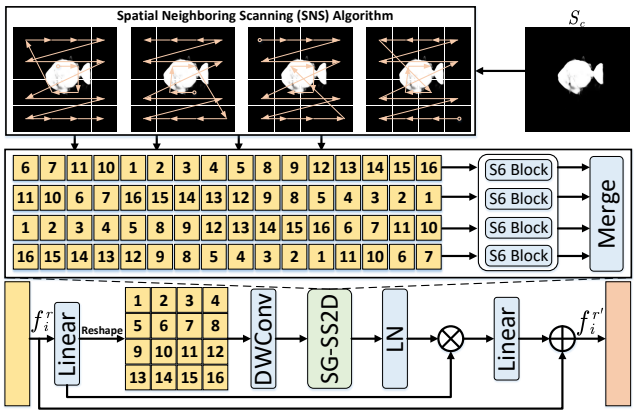

Figure 4. Diagram of the saliency guided Mamba block (SGMB).

3.5. Decoder

As discussed in Sec. 1, two crucial issues remain in designing Mamba-based SOD decoders. To address them, we propose a novel saliency-guided Mamba block (SGMB) and a context-aware upsampling (CAU) method.

3.5.1. Saliency-guided Mamba Block

As shown in Fig. 2, the extracted RGB features f_i^r, where $i \in [1, 2, 3]$, and a coarse saliency map S_c predicted from f_4^r are fed to SGMB, aiming to enhance the RGB features. As emphasized in Sec. 1, maintaining spatial continuity of salient patches within 1D sequences is crucial for accurate prediction by SSM. To this end, we design a spatial neighboring scanning (SNS) algorithm that flattens 2D feature maps into 1D sequences while preserving spatial continuity of salient patches, as illustrated in Fig. 4. Specifically, we transfer maintaining spatial continuity of salient patches to a shortest path traversal problem of salient patches. In other words, during scanning, the next salient patch to be scanned should be spatially close to the current salient patch. In order to reduce computational complexity and keep the traversal path as short as possible, SNS scans each salient patch through an approximate shortest path. The core process of SNS is detailed in Algorithm 1.

The input is S_c, where each salient patch is assigned an index representing its position in the map. The output is a list I_s that stores the indexes of all salient patches, reflecting the scanning path of salient regions. Starting from the first

Algorithm 1 Core process of SNS

Input: 2D coarse saliency map S_c with shape = (h, w)
Output: 1D array I_s that stores the indexes of all salient patches

Initialize the current scanning row of S_c as the first row ($cur = 1$), the scanning direction of cur from left to right ($dir = l \rightarrow r$) and 1D array I_s as empty ($I_s = \varnothing$).
while $cur <= h$ **do**
  Scan the salient patches in the current row according to the direction dir, and append their indexes to I_s.
  Calculate the distances between the last salient patch in I_s and both the leftmost ($dist_{left}$) and rightmost ($dist_{right}$) salient patches in the next row.
  if $dist_{left} <= dist_{right}$ **then**
    Set the scanning direction from left to right ($l \rightarrow r$).
  else
    Set the scanning direction from right to left ($r \rightarrow l$).
  end if
  Jump to the next row($cur = cur + 1$).
end while
Return: I_s

Figure 5. Scanning paths of salient regions generated by SNS.

row, SNS scans all salient patches row by row. Since salient patches within the same row are almost continuous, we can approximate that two adjacent patches within the same row are the closest to each other. Thus, we can scan the salient patches within each row either from left to right or right to left. After scanning the current row, the algorithm moves to the next row. To maintain spatial continuity and minimize computation time, we compare the distance between the last salient patch in the current row and both the leftmost and rightmost salient patches in the next row. The patch with the smaller distance is selected as the starting patch in the next row. These steps are repeated until all rows have been scanned, and the final I_s is generated. Once I_s is obtained, we store the indexes of all non-salient patches sequentially in a list I_{ns}, and then concatenate it to I_s, generating a complete scanning path for 2D feature maps. To enhance the robustness of SNS, we generate three variants of the scanning path by altering the directions: 1) concatenate I_s to I_{ns}; 2) reverse I_s, I_{ns} and concatenate I_{ns} to I_s; 3) reverse I_s, I_{ns} and concatenate I_s to I_{ns}. These scanning paths are then applied to the RGB features, flattening them into 1D sequences. The remaining steps of SGMB are similar to those in the VSS block. Finally, the input RGB features f_i^r are enhanced into high-quality features $f_i^{r\prime}$.

Discussions of SNS. (1) Compared to the "S" pattern in Fig. 1, our SNS requires extra computation to determine the scanning direction for each row to preserve spatial continuity. However, in certain cases, SNS could generate the similar scanning paths of salient regions as the "S" pattern. Thus, to confirm the necessity of SNS, we count the number of images where scanning paths of salient regions are different from the "S" pattern, and its proportion across all images is found to be $\sim$31%. Besides, some visual examples are provided in Fig. 5. **(2)** Notably, from another perspective, the proposed SNS can be deemed as a novel approach to inject information of a prior map into the processed features, by changing patch orders according to the map. Such an approach could provide insights for future designs of Mamba-based infrastructures.

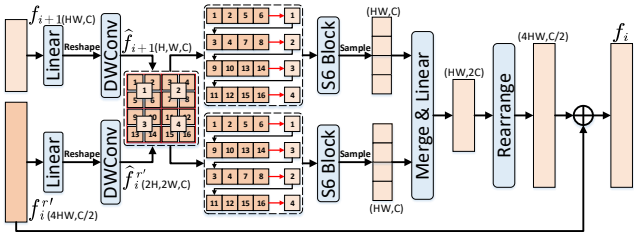

Figure 6. Diagram of the context-aware upsampling (CAU) method.

3.5.2. Context-aware Upsampling Method

Previous upsampling methods lack learnability and fail to model the contextual dependencies between hierarchical features, leading to misalignment during feature fusion. To tackle this, we propose a learnable context-aware upsampling (CAU) method, and integrate it into the decoder.

Fig. 6 shows the diagram of CAU. Firstly, two input features $f_{i+1} \in \mathbb{R}^{HW \times C}$ and $f_i^{r\prime} \in \mathbb{R}^{4HW \times C/2}$, where $i \in [1, 2, 3]$, are processed by a Linear, a Reshape and a DWConv, yielding $\hat{f}_{i+1} \in \mathbb{R}^{H \times W \times C}$ and $\hat{f}_i^{r\prime} \in \mathbb{R}^{2H \times 2W \times C}$. To align $\hat{f}_{i+1}$ with $\hat{f}_i^{r\prime}$ after upsampling, we propose leveraging the patch information of $\hat{f}_i^{r\prime}$ to guide the upsampling process. Due to the neighborhood correlation of patches between hierarchical features, each patch in $\hat{f}_{i+1}$ can be associated with four most relevant patches within $\hat{f}_i^{r\prime}$. Based on this, we can model the contextual dependencies between $\hat{f}_{i+1}$ and $\hat{f}_i^{r\prime}$. Specifically, we first group the patches of $\hat{f}_i^{r\prime}$ using 2×2 windows. Then, we sample the patches from $\hat{f}_{i+1}$ one by one, and sequentially pair them with the patch groups of $\hat{f}_i^{r\prime}$. Next, we concatenate the paired subsequences into a long sequence, and input it to an S6 block. By exploiting the causal prediction capabilities of the S6 block, each patch from $\hat{f}_{i+1}$ can progressively simulates the feature distribution of its corresponding patch group from $\hat{f}_i^{r\prime}$. To enhance this simulation process, we alter the connection order of the paired subsequences to generate a new long sequence, and input it to the other S6 block. Afterward, we sample the patches belonging to $\hat{f}_{i+1}$ from the processed long sequences, and restore them to the same order, yielding two features shaped as $\mathbb{R}^{HW \times C}$. Then they are merged and processed by a Linear layer to generate a new feature shaped as $\mathbb{R}^{HW \times 2C}$. To reorganize its feature distribution, we use a rearrange function to expand its length fourfold and reduce its channels to a quarter of the original, obtaining an upsam-

Table 1. Quantitative comparison of our *Samba* against other SOTA RGB SOD methods on five benchmark datasets. "-" indicates the result is not available. "↑" denotes that the larger value is better. The best two results are stressed in red and blue.

Method	Params (M)	MACs (G)	DUTS [71]			DUT-O [77]			HKU-IS [34]			PASCAL-S [38]			ECSSD [75]		
			$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$
CNN-based																	
GateNet-R [86]	128.63	162.22	0.891	0.874	0.932	0.840	0.782	0.878	0.921	0.926	0.959	0.863	0.836	0.886	0.924	0.935	0.955
CSF-R2 [15]	36.53	18.96	0.890	0.869	0.929	0.838	0.775	0.869	-	-	-	0.863	0.839	0.885	0.931	0.942	0.960
EDN [73]	42.85	20.41	0.892	0.893	0.933	0.849	0.821	0.884	0.924	0.940	0.963	0.864	0.879	0.907	0.927	0.950	0.957
ICON-R [93]	33.09	20.91	0.890	0.876	0.931	0.845	0.799	0.884	0.920	0.931	0.960	0.862	0.844	0.888	0.928	0.943	0.960
MENet [72]	27.83	94.62	0.905	0.895	0.943	0.850	0.792	0.879	0.927	0.939	0.965	0.871	0.848	0.892	0.927	0.938	0.956
Transformer-based																	
EBM [79]	118.96	53.38	0.909	0.900	0.949	0.858	0.817	0.900	0.930	0.943	0.971	0.877	0.856	0.899	0.941	0.954	0.972
ICON-S [93]	94.30	52.59	0.917	0.911	0.960	0.869	0.830	0.906	0.936	0.947	0.974	0.885	0.860	0.903	0.941	0.954	0.971
BBRF [48]	74.40	48.60	0.908	0.905	0.951	0.855	0.820	0.898	0.935	0.946	0.936	0.871	0.884	0.925	0.939	0.957	0.972
VST-S ++ [40]	74.90	32.73	0.909	0.897	0.947	0.859	0.813	0.890	0.932	0.941	0.969	0.880	0.859	0.901	0.939	0.951	0.969
VSCode-S [47]	74.72	93.76	0.926	0.922	0.960	0.877	0.840	0.912	0.940	0.951	0.974	0.887	0.864	0.904	0.949	0.959	0.974
Samba	49.59	46.68	0.932	0.930	0.966	0.889	0.859	0.922	0.945	0.956	0.978	0.892	0.896	0.931	0.953	0.965	0.978

Table 2. Quantitative comparison of our *Samba* against other SOTA RGB-D SOD methods on five benchmark datasets.

Method	Params (M)	MACs (G)	NJUD [30]			NLPR [56]			SIP [9]			STERE [54]			DUTLF-D [58]		
			$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$
CNN-based																	
BBSNet [11]	49.77	31.20	0.921	0.919	0.949	0.931	0.918	0.961	0.879	0.884	0.922	0.908	0.903	0.942	0.882	0.870	0.912
JL-DCF [12]	143.52	211.06	0.877	0.892	0.941	0.931	0.918	0.965	0.885	0.894	0.931	0.900	0.895	0.942	0.894	0.891	0.927
SP-Net [88]	67.88	175.29	0.925	0.928	0.957	0.927	0.919	0.962	0.894	0.904	0.933	0.907	0.906	0.949	0.895	0.899	0.933
DCF [27]	53.92	108.60	0.904	0.905	0.943	0.922	0.910	0.957	0.874	0.886	0.922	0.906	0.904	0.948	0.925	0.930	0.956
SPSN [32]	-	-	0.918	0.921	0.952	0.923	0.912	0.960	0.892	0.900	0.936	0.907	0.902	0.945	-	-	-
Transformer-based																	
SwinNet-B [45]	199.18	122.20	0.920	0.924	0.956	0.941	0.936	0.974	0.911	0.927	0.950	0.919	0.918	0.956	0.918	0.920	0.949
CATNet [63]	262.73	172.06	0.932	0.937	0.960	0.938	0.934	0.971	0.910	0.928	0.951	0.920	0.922	0.958	0.952	0.958	0.975
VST-S ++ [40]	143.15	45.41	0.928	0.928	0.957	0.935	0.925	0.964	0.904	0.918	0.946	0.921	0.916	0.954	0.945	0.950	0.969
CPNet [23]	216.50	129.34	0.935	0.941	0.963	0.940	0.936	0.971	0.907	0.927	0.946	0.920	0.922	0.960	0.951	0.959	0.974
VSCode-S [47]	74.72	93.76	0.944	0.949	0.970	0.941	0.932	0.968	0.924	0.942	0.958	0.931	0.928	0.958	0.960	0.967	0.980
Samba	54.94	71.64	0.949	0.956	0.975	0.947	0.941	0.976	0.931	0.949	0.966	0.935	0.933	0.963	0.956	0.964	0.976

pled feature shaped as $\mathbb{R}^{4HW\times C/2}$. Finally, the original $f_i^{r\prime}$ is added to the upsampled feature for feature aggregation.

3.5.3. VSS Decoder Layers

We implement VSS decoder layers based on VSS blocks, aiming to decode the aggregated features from CAU. To explore inter-channel dependencies, we introduce a channel attention mechanism (CAM) [22] following SS2D, forming our VSS decoder layers. The process of these layers mainly follows a $LN \rightarrow Linear \rightarrow DWConv \rightarrow SS2D \rightarrow CAM \rightarrow LN \rightarrow Linear$ flow with a residual connection.

4. Experiments

4.1. Datasets and Metrics

For **RGB SOD**, we evaluate *Samba* on five commonly used benchmark datasets, i.e., DUTS [71], DUT-O [77], HKU-IS [34], PASCAL-S [38] and ECSSD [75]. As for **RGB-D SOD**, we use five benchmark datasets, including NJUD [30], NLPR [56], SIP [9], STERE [54] and DUTLF-D [58]. Regarding **RGB-T SOD**, we employ three benchmark datasets: VT821 [69], VT1000 [67] and VT5000 [66]. For **VSOD**, we utilize five widely used benchmark datasets: DAVIS [57], DAVSOD-easy [10], FBMS [55], SegV2 [33] and VOS [37]. In terms of **RGB-D VSOD**, three public datasets are considered, including RDVS [53], DVisal [36] and ViDSOD [39]. We adopt three saliency metrics to evaluate model performance, i.e., structure-measure (S_m) [8], maximum F-measure (F_m) [1] and maximum enhanced-alignment measure (E_m) [5]. To assess model computational complexity and model size, we also report the multiply accumulate operations (MACs) and the number of parameters (Params).

4.2. Implementation Details

Our *Samba* is implemented in PyTorch trained on an NVIDIA 4090 GPU. Following previous works, we have arranged the training sets for each task as follows: the training set of DUTS for **RGB SOD**, the training sets of NJUD, NLPR and DUTLF-D for **RGB-D SOD**, the training set of VT5000 for **RGB-T SOD**, the training sets of DAVIS and DAVSOD for **VSOD**. Due to the lack of a benchmark training set for **RGB-D VSOD**, we train and test *Samba* on RDVS, DVisal and ViDSOD separately. In the training process, we adopt AdamW optimizer with an initial learning rate of $1e-4$ and the batch size of 2. All input images are uniformly resized to 448 × 448 for training and testing, and are also augmented using various strategies like random flipping, random cropping and random rotating during training. The model converges after 30 training epochs.

4.3. Comparison with State-of-the-Art Methods

Quantitative Evaluation. Since our *Samba* is a unified model to handle general SOD tasks, we present comparative experiments of *Samba* against existing SOTA methods across five SOD tasks, including 10 models for RGB SOD, 10 models for RGB-D SOD, 10 models for VSOD, 10 models for RGB-T SOD, 3 models for RGB-D VSOD, as shown in Table 1, 2, 3, 4 and 5. The comprehensive results illustrate that

Table 3. Quantitative comparison of our *Samba* against other SOTA VSOD methods on five benchmark datasets.

Method	Params (M)	MACs (G)	DAVIS [57]			DAVSOD-easy [10]			FBMS [55]			SegV2 [33]			VOS [37]		
			$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$
CNN-based																	
MGAN [35]	91.51	123.57	0.913	0.894	0.965	0.740	0.611	0.778	0.909	0.903	0.946	0.902	0.869	0.950	0.797	0.725	0.829
PCSA [19]	-	-	0.900	0.877	0.960	0.725	0.590	0.759	0.872	0.844	0.917	0.886	0.848	0.938	0.802	0.699	0.816
FSNet [26]	83.41	35.32	0.922	0.909	0.972	0.760	0.637	0.796	0.875	0.867	0.918	0.849	0.773	0.920	0.678	0.621	0.755
DCFNet [80]	69.56	93.27	0.914	0.899	0.970	0.729	0.612	0.781	0.883	0.853	0.910	0.903	0.870	0.953	0.838	0.773	0.861
UGPL [59]	-	-	0.911	0.895	0.968	0.732	0.602	0.771	0.897	0.884	0.939	0.867	0.828	0.938	0.751	0.685	0.811
MMNet [85]	50.81	82.63	0.911	0.895	0.968	0.732	0.602	0.771	0.897	0.884	0.939	0.867	0.828	0.938	0.751	0.685	0.811
Transformer-based																	
MGTNet [52]	150.91	265.21	0.925	0.919	0.976	0.765	0.653	0.800	0.900	0.881	0.929	0.903	0.861	0.946	0.814	0.727	0.819
UFO [21]	55.92	248.80	0.918	0.906	0.978	0.747	0.626	0.799	0.858	0.868	0.911	0.888	0.850	0.951	-	-	-
CoSTFormer [41]	-	-	0.923	0.906	0.978	0.779	0.667	0.819	0.869	0.861	0.913	0.874	0.813	0.943	0.791	0.708	0.811
VSCode-S [47]	74.72	93.76	0.936	0.922	0.973	0.800	0.710	0.835	0.905	0.902	0.939	0.946	0.937	0.984	-	-	-
Samba	54.94	71.64	0.943	0.936	0.985	0.813	0.734	0.856	0.925	0.922	0.954	0.943	0.938	0.987	0.870	0.820	0.898

Table 4. Quantitative comparison of our *Samba* against other SOTA RGB-T SOD methods on three benchmark datasets.

Method		CNN-based								Transformer-based		
		MIDD [65]	ECFFNet [89]	CGFNet [70]	CSRNet [25]	MGAI [62]	TNet [4]	CGMDR [2]	SPNet [82]	SwinNet-B [45]	VSCode-S [47]	*Samba*
Params (M)		52.43	-	69.92	1.01	87.09	87.04	-	104.03	199.18	74.72	54.94
MACs (G)		217.13	-	382.63	5.76	78.37	54.90	-	67.59	122.20	93.76	71.64
VT821 [69]	$S_m \uparrow$	0.871	0.877	0.881	0.885	0.891	0.899	0.894	0.913	0.904	0.926	0.934
	$F_m \uparrow$	0.847	0.835	0.866	0.855	0.870	0.885	0.872	0.900	0.877	0.910	0.927
	$E_m \uparrow$	0.916	0.911	0.920	0.920	0.933	0.936	0.932	0.949	0.937	0.954	0.965
VT1000 [67]	$S_m \uparrow$	0.916	0.924	0.923	0.919	0.929	0.929	0.931	0.941	0.938	0.952	0.953
	$F_m \uparrow$	0.904	0.919	0.923	0.901	0.921	0.921	0.927	0.943	0.933	0.947	0.956
	$E_m \uparrow$	0.956	0.959	0.959	0.952	0.965	0.965	0.966	0.975	0.974	0.981	0.983
VT5000 [66]	$S_m \uparrow$	0.868	0.876	0.883	0.868	0.884	0.895	0.896	0.914	0.912	0.925	0.928
	$F_m \uparrow$	0.834	0.850	0.852	0.821	0.846	0.864	0.877	0.905	0.885	0.900	0.919
	$E_m \uparrow$	0.919	0.922	0.926	0.912	0.930	0.936	0.939	0.954	0.944	0.959	0.963

Table 5. Quantitative comparison of our *Samba* against other SOTA RGB-D VSOD methods on three public datasets.

Method		CNN-based			*Samba*
		DCTNet+ [53]	DVSOD [36]	ATFNet [39]	
Params (M)		90.69	97.34	124.07	60.28
MACs (G)		117.94	276.46	54.36	96.60
RDVS [53]	$S_m \uparrow$	0.869	0.689	0.741	0.883
	$F_m \uparrow$	0.814	0.574	0.592	0.834
	$E_m \uparrow$	0.914	0.733	0.785	0.936
DVisal [36]	$S_m \uparrow$	0.814	0.729	0.723	0.847
	$F_m \uparrow$	0.807	0.648	0.659	0.825
	$E_m \uparrow$	0.909	0.813	0.809	0.914
ViDSOD [39]	$S_m \uparrow$	0.877	0.770	0.875	0.923
	$F_m \uparrow$	0.820	0.687	0.832	0.895
	$E_m \uparrow$	0.901	0.846	0.911	0.944

Samba outperforms existing SOTA CNN- and transformer-based SOD models across 21 datasets, with a comparable number of Params and relatively low MACs, demonstrating the superior performance of *Samba*. Specifically, for RGB SOD, the Params and MACs of *Samba* are lower than those of transformer-based methods (except VST-S++ [40]), while *Samba* also achieves best results on the given datasets. Although some CNN-based methods (ICON-R [93], EDN [73] and CSF-R2 [15]) are more efficient than our *Samba*, their performance is substantially inferior to *Samba*. Regarding RGB-D/T SOD, VSOD and RGB-D VSOD, most methods (except BBSNet [11], VST-S++ [40], FSNet [26], SPNet [82], TNet [4], CSRNet [25], ATFNet [39]) exhibit higher computational complexity than *Samba*, regardless of whether they are CNN- or transformer-based. Nevertheless, *Samba* still outperforms these methods on the given datasets. It should be noted that VSCode-S has received joint training data from all SOD sub-tasks, since it aims at a general prompt-based scheme, while *Samba* is only trained on individual SOD sub-tasks separately. In this regard, VSCode-S has been fed with more training data than *Samba*.

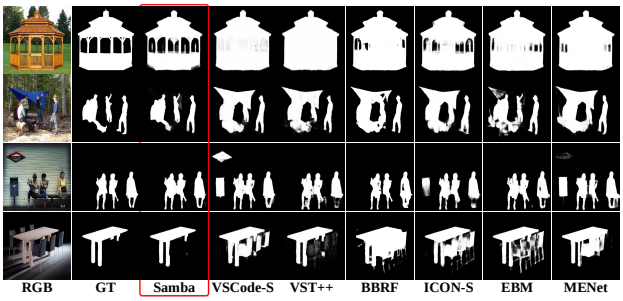

Figure 7. Visual comparison against SOTA RGB SOD methods.

Qualitative Evaluation. To clearly demonstrate the superiority of our *Samba*, we display visual comparison results among the top-performing RGB SOD models in Fig. 7. The 1^{st} row showcases a large object with hollow parts. In comparison to other models, *Samba* can accurately detect hollow parts and produce more reliable results. In the 2^{nd} and 3^{rd} rows, we present two scenes with multiple salient objects. From the comparison results, it can be observed that *Samba* effectively locates all salient objects and segments them more accurately than the other models. The last row depicts a scene with cluttered backgrounds and object occlusion, where *Samba* successfully detects the salient object while other models misidentify non-salient regions.

4.4. Ablation Study

To verify the relative contribution of different components in *Samba*, we conduct thorough ablation studies by removing

Table 6. Ablation study of *Samba* on three RGB SOD, three RGB-D SOD and one RGB-D VSOD datasets. **Bolded** results are the best.

Settings	DUTS[71]			ECSSD[75]			DUT-O[77]			NJUD[30]			NLPR[56]			DUTLF-D[58]			RDVS[53]		
	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$
Ours	**0.932**	**0.930**	**0.966**	**0.889**	**0.859**	**0.922**	**0.953**	**0.965**	**0.978**	**0.949**	**0.956**	**0.975**	**0.947**	**0.941**	**0.976**	**0.956**	**0.964**	**0.976**	**0.883**	**0.834**	**0.936**
A1	0.922	0.916	0.950	0.881	0.843	0.909	0.946	0.954	0.966	0.940	0.941	0.961	0.942	0.932	0.965	0.947	0.952	0.963	0.876	0.822	0.929
A2	0.926	0.919	0.952	0.882	0.846	0.914	0.946	0.952	0.964	0.941	0.945	0.966	0.945	0.933	0.962	0.949	0.957	0.968	0.878	0.827	0.930
A3	0.929	0.924	0.961	0.884	0.852	0.917	0.950	0.961	0.973	0.945	0.949	0.970	0.944	0.937	0.974	0.949	0.959	0.971	0.880	0.829	0.931
A4	0.928	0.927	0.963	0.886	0.853	0.917	0.947	0.960	0.971	0.943	0.948	0.965	0.943	0.938	0.972	0.947	0.956	0.972	0.879	0.831	0.932
A5	0.930	0.928	0.963	0.886	0.857	0.918	0.951	0.962	0.973	0.946	0.952	0.972	0.945	0.940	0.973	0.949	0.961	0.972	0.881	0.831	0.933
A6	0.928	0.926	0.963	0.886	0.856	0.918	0.948	0.960	0.971	0.947	0.954	0.973	0.945	0.940	0.972	0.952	0.962	0.975	0.880	0.830	0.933
B1	0.927	0.923	0.960	0.887	0.856	0.912	0.952	0.961	0.976	0.941	0.952	0.966	0.942	0.939	0.972	0.953	0.957	0.969	0.879	0.828	0.932
B2	0.921	0.918	0.952	0.877	0.848	0.905	0.949	0.955	0.970	0.936	0.944	0.962	0.939	0.933	0.965	0.947	0.954	0.961	0.874	0.819	0.928
B3	0.923	0.924	0.956	0.884	0.855	0.913	0.946	0.959	0.968	0.937	0.946	0.966	0.938	0.933	0.963	0.950	0.955	0.966	0.878	0.826	0.933
B4	0.931	0.928	0.964	0.887	0.857	0.921	0.951	0.965	0.974	0.947	0.955	0.972	0.946	0.939	0.971	0.955	0.963	0.974	0.882	0.834	0.934
B5	0.929	0.926	0.964	0.888	0.854	0.920	0.950	0.964	0.976	0.948	0.953	0.971	0.944	0.936	0.973	0.955	0.962	0.971	0.879	0.832	0.931
B6	0.926	0.922	0.963	0.888	0.853	0.916	0.949	0.961	0.974	0.945	0.952	0.971	0.940	0.934	0.972	0.952	0.959	0.972	0.877	0.833	0.928
C1	0.932	0.930	0.966	0.889	0.859	0.922	0.953	0.965	0.978	0.944	0.954	0.970	0.945	0.936	0.973	0.952	0.961	0.969	0.878	0.831	0.926
C2	0.929	0.926	0.965	0.886	0.857	0.918	0.951	0.962	0.973	0.946	0.953	0.973	0.944	0.938	0.972	0.953	0.963	0.972	0.881	0.833	0.932

or replacing them from *Samba*. As shown in Table 6, we perform a range of experiments on three RGB SOD (DUTS, ECSSD, DUT-O), three RGB-D SOD (NJUD, NLPR, DUTLF-D) and one RGB-D VSOD datasets (RDVS).

Effectiveness of SGMB. To validate the effectiveness of SGMB, we first delete the SGMB module, denoted as variant "A1" in Table 6, and then utilize an SS2D module to replace the SG-SS2D module of SGMB, denoted as variant "A2". From the comparison results, it is evident that our model performs better than "A1" and "A2", which indicates the significant contribution of SGMB in boosting detection performance. Since the core idea of SGMB lies in the SNS algorithm, which is used to maintain spatial continuity of salient patches. To evaluate its contribution, we apply three other scanning manners to salient regions, i.e., Fig. 1 (a), (b) and (c), which fail to preserve spatial continuity of salient patches. We denote these evaluations as "A3", "A4" and "A5". Compared to these variants, our model consistently outperforms them on the evaluated datasets, highlighting the importance of preserving spatial continuity of salient patches. In the SNS algorithm, we generate three variants of the scanning path by changing the directions to enhance the robustness of SNS. To validate the contribution of the three paths, we create three copies of the initial scanning path to replace them, denoted as "A6". The comparison results between "Ours" and "A6" demonstrate the effectiveness of using multiple paths with varying directions.

Effectiveness of CAU. The proposed CAU module achieves learnable upsampling by modeling contextual dependencies between hierarchical features, which facilitates the alignment and aggregations of these features. To evaluate its effectiveness, we compare it with three other upsampling methods. Firstly, we utilize nearest-neighbor interpolation to replace the upsampling operation in the CAU module, denoted as variant "B1". Besides, we investigate two other learnable upsampling methods: DUpsampling [64] and DySample [42], and apply them separately for feature upsampling, denoted as "B2" and "B3". Comparing "B1", "B2" and "B3" with our full model, it is clear that CAU outperforms all alternative upsampling methods. In the CAU module, we introduce a novel patch pairing and ordering scheme, which plays a central role in the upsampling process. To validate the effectiveness of this design, we shift the original pairing sequences by one patch, three patches, and five patches, resulting in variants "B4", "B5" and "B6". The performance of these variants shows a clear degradation with the increased shifting of the original pairing order, indicating the effectiveness of the proposed patch pairing and ordering scheme.

Effectiveness of MFM. In order to validate the proposed MFM module, we replace it with a simple concatenation followed by a convolution, denoted as variant "C1". Besides, we re-implement the convertor using the RFM module proposed in [53], denoted as variant "C2". The comparison results clearly demonstrate the superiority of MFM in facilitating the interaction of multi-modal information.

5. Conclusion

In this paper, we are the first to adapt state space models to SOD tasks, proposing a new unified framework based on the pure Mamba architecture, named saliency Mamba (*Samba*), which can flexibly handle general SOD tasks. We identify spatial continuity of salient patches within scanning sequences, and propose a novel saliency-guided Mamba block (SGMB). Central to SGMB is a spatial neighboring scanning (SNS) algorithm, which dynamically adjusts scanning directions to maintain spatial continuity of salient patches. Furthermore, we propose a context-aware upsampling (CAU) to promote hierarchical feature alignment and aggregations by modeling contextual dependencies. Extensive experimental results demonstrate that our *Samba* outperforms existing SOTA CNN- and transformer-based models across five SOD tasks on 21 datasets, with lower computational cost.

Acknowledgments. This work was supported by the National Natural Science Foundation of China, under No. 62176169, 62301310, 62176170, and the Sichuan Science and Technology Program (2025ZNSFSC0469, 2024NSFSC1426).

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604. IEEE, 2009. 6

[2] Gang Chen, Feng Shao, Xiongli Chai, Hangwei Chen, Qiuping Jiang, Xiangchao Meng, and Yo-Sung Ho. Cgmdrnet: Cross-guided modality difference reduction network for rgb-t salient object detection. *IEEE TCSVT*, 32(9):6308–6323, 2022. 2, 7

[3] Qian Chen, Zhenxi Zhang, Yanye Lu, Keren Fu, and Qijun Zhao. 3-d convolutional neural networks for rgb-d salient object detection and beyond. *IEEE TNNLS*, 35(3):4309–4323, 2024. 2

[4] Runmin Cong, Kepu Zhang, Chen Zhang, Feng Zheng, Yao Zhao, Qingming Huang, and Sam Kwong. Does thermal really always matter for rgb-t salient object detection? *IEEE TMM*, 25:6971–6982, 2022. 1, 2, 7

[5] Yang Cao Bo Ren Ming-Ming Cheng Ali Borji Deng-Ping Fan, Cheng Gong. Enhanced-alignment measure for binary foreground map evaluation. In *IJCAI*, 2018. 6

[6] Wenhao Dong, Haodong Zhu, Shaohui Lin, Xiaoyan Luo, Yunhang Shen, Xuhui Liu, Juan Zhang, Guodong Guo, and Baochang Zhang. Fusion-mamba for cross-modality object detection. *arXiv preprint arXiv:2404.09146*, 2024. 1, 3

[7] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 4

[8] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. 6

[9] Deng-Ping Fan, Zheng Lin, Zhao Zhang, Menglong Zhu, and Ming-Ming Cheng. Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks. *IEEE TNNLS*, 32(5):2075–2089, 2020. 6

[10] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564, 2019. 6, 7

[11] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network. In *ECCV*, pages 275–292. Springer, 2020. 6, 7

[12] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *CVPR*, pages 3052–3062, 2020. 2, 6

[13] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, Qijun Zhao, Jianbing Shen, and Ce Zhu. Siamese network for rgb-d salient object detection and beyond. *IEEE TPAMI*, 44(9):5541–5559, 2022. 2

[14] Keren Fu, Yao Jiang, Ge-Peng Ji, Tao Zhou, Qijun Zhao, and Deng-Ping Fan. Light field salient object detection: A review and benchmark. *Computational Visual Media*, 8(4):509–534, 2022. 1

[15] Shang-Hua Gao, Yong-Qiang Tan, Ming-Ming Cheng, Chengze Lu, Yunpeng Chen, and Shuicheng Yan. Highly efficient salient object detection with 100k parameters. In *ECCV*, pages 702–721. Springer, 2020. 6, 7

[16] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 3, 4

[17] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 3

[18] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *NeurIPS*, volume 34, pages 572–585, 2021. 3

[19] Yuchao Gu, Lijuan Wang, Ziqin Wang, Yun Liu, Ming-Ming Cheng, and Shao-Ping Lu. Pyramid constrained self-attention network for fast video salient object detection. In *AAAI*, volume 34, pages 10869–10876, 2020. 2, 7

[20] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *ECCV*, pages 222–241. Springer, 2024. 2

[21] Ruohao Guo, Xianghua Ying, Yanyu Qi, and Liao Qu. Unitr: A unified transformer-based framework for co-object and multi-modal saliency detection. *IEEE TMM*, 2024. 1, 2, 7

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 6

[23] Xihang Hu, Fuming Sun, Jing Sun, Fasheng Wang, and Haojie Li. Cross-modal fusion and progressive decoding network for rgb-d salient object detection. *IJCV*, pages 1–19, 2024. 1, 2, 6

[24] Nianchang Huang, Yang Yang, Dingwen Zhang, Qiang Zhang, and Jungong Han. Employing bilinear fusion and saliency prior information for rgb-d salient object detection. *IEEE TMM*, 24:1651–1664, 2021. 2

[25] Fushuo Huo, Xuegui Zhu, Lei Zhang, Qifeng Liu, and Yu Shu. Efficient context-guided stacked refinement network for rgb-t salient object detection. *IEEE TCSVT*, 32(5):3111–3124, 2021. 2, 7

[26] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *ICCV*, pages 4922–4933, 2021. 2, 7

[27] Wei Ji, Jingjing Li, Shuang Yu, Miao Zhang, Yongri Piao, Shunyu Yao, Qi Bi, Kai Ma, Yefeng Zheng, Huchuan Lu, et al. Calibrated rgb-d salient object detection. In *CVPR*, pages 9471–9481, 2021. 6

[28] Yao Jiang, Xin Li, Keren Fu, and Qijun Zhao. Transformer-based light field salient object detection and its application to autofocus. *IEEE TIP*, 33:6647–6659, 2024. 1

[29] Yao Jiang, Xinyu Yan, Ge-Peng Ji, Keren Fu, Meijun Sun, Huan Xiong, Deng-Ping Fan, and Fahad Shahbaz Khan. Effectiveness assessment of recent large vision-language models. *Visual Intelligence*, 2(1):17, 2024. 1

[30] Ran Ju, Ling Ge, Wenjing Geng, Tongwei Ren, and Gangshan Wu. Depth saliency based on anisotropic center-surround difference. In *ICIP*, pages 1115–1119. IEEE, 2014. 6, 8

[31] Yun Yi Ke and Takahiro Tsubono. Recursive contour-saliency blending network for accurate salient object detection. In *WACV*, pages 2940–2950, 2022. 2

[32] Minhyeok Lee, Chaewon Park, Suhwan Cho, and Sangyoun Lee. Spsn: Superpixel prototype sampling network for rgb-d salient object detection. In *ECCV*, pages 630–647. Springer, 2022. 2, 6

[33] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and

James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 6, 7

[34] Guanbin Li and Yizhou Yu. Visual saliency based on multi-scale deep features. In *CVPR*, pages 5455–5463, 2015. 6

[35] Haofeng Li, Guanqi Chen, Guanbin Li, and Yizhou Yu. Motion guided attention for video salient object detection. In *ICCV*, pages 7274–7283, 2019. 2, 7

[36] Jingjing Li, Wei Ji, Size Wang, Wenbo Li, et al. Dvsod: Rgb-d video salient object detection. In *NeurIPS*, volume 36, 2024. 1, 2, 6, 7

[37] Jia Li, Changqun Xia, and Xiaowu Chen. A benchmark dataset and saliency-guided stacked autoencoders for video-based salient object detection. *IEEE TIP*, 27(1):349–364, 2017. 6, 7

[38] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014. 6

[39] Junhao Lin, Lei Zhu, Jiaxing Shen, Huazhu Fu, Qing Zhang, and Liansheng Wang. Vidsod-100: A new dataset and a baseline model for rgb-d video salient object detection. *IJCV*, pages 1–19, 2024. 1, 2, 6, 7

[40] Nian Liu, Ziyang Luo, Ni Zhang, and Junwei Han. Vst++: Efficient and stronger visual saliency transformer. *IEEE TPAMI*, 2024. 1, 2, 6, 7

[41] Nian Liu, Kepan Nan, Wangbo Zhao, Xiwen Yao, and Junwei Han. Learning complementary spatial–temporal transformer for video salient object detection. *IEEE TNNLS*, 2023. 2, 7

[42] Wenze Liu, Hao Lu, Hongtao Fu, and Zhiguo Cao. Learning to upsample by learning to sample. In *ICCV*, pages 6027–6037, 2023. 2, 8

[43] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024. 1, 2, 3

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 1

[45] Zhengyi Liu, Yacheng Tan, Qian He, and Yun Xiao. Swinnet: Swin transformer drives edge-aware rgb-d and rgb-t salient object detection. *IEEE TCSVT*, 32(7):4486–4497, 2021. 6, 7

[46] Yukang Lu, Dingyao Min, Keren Fu, and Qijun Zhao. Depth-cooperated trimodal network for video salient object detection. In *ICIP*, pages 116–120. IEEE, 2022. 2

[47] Ziyang Luo, Nian Liu, Wangbo Zhao, Xuguang Yang, Dingwen Zhang, Deng-Ping Fan, Fahad Khan, and Junwei Han. Vscode: General visual salient and camouflaged object detection with 2d prompt learning. In *CVPR*, pages 17169–17180, 2024. 6, 7

[48] Mingcan Ma, Changqun Xia, Chenxi Xie, Xiaowu Chen, and Jia Li. Boosting broader receptive fields for salient object detection. *IEEE TIP*, 32:1026–1038, 2023. 2, 6

[49] Xianping Ma, Xiaokang Zhang, and Man-On Pun. Rs 3 mamba: Visual state space model for remote sensing image semantic segmentation. *IEEE GRSL*, 2024. 1, 2

[50] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 1

[51] S Mahdi H Miangoleh, Zoya Bylinskii, Eric Kee, Eli Shechtman, and Yağiz Aksoy. Realistic saliency guided image enhancement. In *CVPR*, pages 186–194, 2023. 1

[52] Dingyao Min, Chao Zhang, Yukang Lu, Keren Fu, and Qijun Zhao. Mutual-guidance transformer-embedding network for video salient object detection. *IEEE SPL*, 29:1674–1678, 2022. 7

[53] Ao Mou, Yukang Lu, Jiahao He, Dingyao Min, Keren Fu, and Qijun Zhao. Salient object detection in rgb-d videos. *IEEE TIP*, 33:6660–6675, 2024. 1, 2, 6, 7, 8

[54] Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. Leveraging stereopsis for saliency analysis. In *CVPR*, pages 454–461. IEEE, 2012. 6

[55] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 36(6):1187–1200, 2013. 6, 7

[56] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgbd salient object detection: A benchmark and algorithms. In *ECCV*, pages 92–109. Springer, 2014. 6, 8

[57] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 6, 7

[58] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *ICCV*, pages 7254–7263, 2019. 6, 8

[59] Yongri Piao, Chenyang Lu, Miao Zhang, and Huchuan Lu. Semi-supervised video salient object detection based on uncertainty-guided pseudo labels. In *NeurIPS*, volume 35, pages 5614–5627, 2022. 2, 7

[60] Yuan Shi, Bin Xia, Xiaoyu Jin, Xing Wang, Tianyu Zhao, Xin Xia, Xuefeng Xiao, and Wenming Yang. Vmambair: Visual state space model for image restoration. *arXiv preprint arXiv:2403.11423*, 2024. 1, 2

[61] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022. 3

[62] Kechen Song, Liming Huang, Aojun Gong, and Yunhui Yan. Multiple graph affinity interactive network and a variable illumination dataset for rgbt image salient object detection. *IEEE TCSVT*, 33(7):3104–3118, 2022. 1, 7

[63] Fuming Sun, Peng Ren, Bowen Yin, Fasheng Wang, and Haojie Li. Catnet: A cascaded and aggregated transformer network for rgb-d salient object detection. *IEEE TMM*, 2023. 1, 2, 6

[64] Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *CVPR*, pages 3126–3135, 2019. 2, 8

[65] Zhengzheng Tu, Zhun Li, Chenglong Li, Yang Lang, and Jin Tang. Multi-interactive dual-decoder for rgb-thermal salient object detection. *IEEE TIP*, 30:5678–5691, 2021. 2, 7

[66] Zhengzheng Tu, Yan Ma, Zhun Li, Chenglong Li, Jieming Xu, and Yongtao Liu. Rgbt salient object detection: A large-scale dataset and benchmark. *IEEE TMM*, 25:4163–4176, 2022. 6, 7

[67] Zhengzheng Tu, Tian Xia, Chenglong Li, Xiaoxiao Wang, Yan Ma, and Jin Tang. Rgb-t image saliency detection via collaborative graph learning. *IEEE TMM*, 22(1):160–173, 2019. 6, 7

[68] Zifu Wan, Yuhao Wang, Silong Yong, Pingping Zhang, Si-

mon Stepputtis, Katia Sycara, and Yaqi Xie. Sigma: Siamese mamba network for multi-modal semantic segmentation. *arXiv preprint arXiv:2404.04256*, 2024. 1, 2, 3

[69] Guizhao Wang, Chenglong Li, Yunpeng Ma, Aihua Zheng, Jin Tang, and Bin Luo. Rgb-t saliency detection benchmark: Dataset, baselines, analysis and a novel approach. In *IGTA*, pages 359–369. Springer, 2018. 6, 7

[70] Jie Wang, Kechen Song, Yanqi Bao, Liming Huang, and Yunhui Yan. Cgfnet: Cross-guided fusion network for rgb-t salient object detection. *IEEE TCSVT*, 32(5):2949–2961, 2021. 7

[71] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017. 6, 8

[72] Yi Wang, Ruili Wang, Xin Fan, Tianzhu Wang, and Xiangjian He. Pixels, regions, and objects: Multiple enhancement for salient object detection. In *CVPR*, pages 10031–10040, 2023. 1, 2, 6

[73] Yu-Huan Wu, Yun Liu, Le Zhang, Ming-Ming Cheng, and Bo Ren. Edn: Salient object detection via extremely-downsampled network. *IEEE TIP*, 31:3125–3136, 2022. 2, 6, 7

[74] Xinyu Xie, Yawen Cui, Tao Tan, Xubin Zheng, and Zitong Yu. Fusionmamba: Dynamic feature enhancement for multimodal image fusion with mamba. *Visual Intelligence*, 2(1):37, 2024. 2

[75] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 6, 8

[76] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*, 2024. 1, 2, 3

[77] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013. 6, 8

[78] Yuhuan Yang, Chaofan Ma, Jiangchao Yao, Zhun Zhong, Ya Zhang, and Yanfeng Wang. Remamber: Referring image segmentation with mamba twister. *arXiv preprint arXiv:2403.17839*, 2024. 1

[79] Jing Zhang, Jianwen Xie, Nick Barnes, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. In *NeurIPS*, volume 34, pages 15448–15463, 2021. 6

[80] Miao Zhang, Jie Liu, Yifei Wang, Yongri Piao, Shunyu Yao, Wei Ji, Jingjing Li, Huchuan Lu, and Zhongxuan Luo. Dynamic context-sensitive filtering network for video salient object detection. In *ICCV*, pages 1553–1563, 2021. 1, 2, 7

[81] Qiang Zhang, Tonglin Xiao, Nianchang Huang, Dingwen Zhang, and Jungong Han. Revisiting feature fusion for rgb-t salient object detection. *IEEE TCSVT*, 31(5):1804–1818, 2020. 2

[82] Zihao Zhang, Jie Wang, and Yahong Han. Saliency prototype for rgb-d and rgb-t salient object detection. In *ACM MM*, pages 3696–3705, 2023. 2, 7

[83] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, pages 8779–8788, 2019. 2

[84] Sijie Zhao, Hao Chen, Xueliang Zhang, Pengfeng Xiao, Lei Bai, and Wanli Ouyang. Rs-mamba for large remote sensing image dense prediction. *arXiv preprint arXiv:2404.02668*, 2024. 1, 2, 3

[85] Xing Zhao, Haoran Liang, Peipei Li, Guodao Sun, Dongdong Zhao, Ronghua Liang, and Xiaofei He. Motion-aware memory network for fast video salient object detection. *IEEE TIP*, 2024. 1, 2, 7

[86] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, pages 35–51. Springer, 2020. 2, 6

[87] Minghang Zhou, Tianyu Li, Chaofan Qiao, Dongyu Xie, Guoqing Wang, Ningjuan Ruan, Lin Mei, and Yang Yang. Dmm: Disparity-guided multispectral mamba for oriented object detection in remote sensing. *arXiv preprint arXiv:2407.08132*, 2024. 1, 3

[88] Tao Zhou, Huazhu Fu, Geng Chen, Yi Zhou, Deng-Ping Fan, and Ling Shao. Specificity-preserving rgb-d saliency detection. In *ICCV*, pages 4681–4691, 2021. 2, 6

[89] Wujie Zhou, Qinling Guo, Jingsheng Lei, Lu Yu, and Jenq-Neng Hwang. Ecffnet: Effective and consistent feature fusion network for rgb-t salient object detection. *IEEE TCSVT*, 32(3):1224–1235, 2021. 7

[90] Zikun Zhou, Wenjie Pei, Xin Li, Hongpeng Wang, Feng Zheng, and Zhenyu He. Saliency-associated object tracking. In *ICCV*, pages 9866–9875, 2021. 1

[91] Enze Zhu, Zhan Chen, Dingkai Wang, Hanru Shi, Xiaoxuan Liu, and Lei Wang. Unetmamba: Efficient unet-like mamba for semantic segmentation of high-resolution remote sensing images. *arXiv preprint arXiv:2408.11545*, 2024. 1, 2, 3

[92] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *ICML*, volume 235, pages 62429–62442. PMLR, 2024. 1, 3

[93] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning. *IEEE TPAMI*, 45(3):3738–3752, 2022. 2, 6, 7