Visual Saliency Transformer

Nian Liu1* Ni Zhang2* Kaiyuan Wan2 Ling Shao1 Junwei Han2†
1Inception Institute of Artificial Intelligence 2Northwestern Polytechnical University
{liunian228, nnizhang.1995, kaiyuan.wan0106, junweihan2010}@gmail.com, ling.shao@ieee.org

Abstract

Existing state-of-the-art saliency detection methods heavily rely on CNN-based architectures. Alternatively, we rethink this task from a convolution-free sequence-to-sequence perspective and predict saliency by modeling long-range dependencies, which can not be achieved by convolution. Specifically, we develop a novel unified model based on a pure transformer, namely, Visual Saliency Transformer (VST), for both RGB and RGB-D salient object detection (SOD). It takes image patches as inputs and leverages the transformer to propagate global contexts among image patches. Unlike conventional architectures used in Vision Transformer (ViT), we leverage multi-level token fusion and propose a new token upsampling method under the transformer framework to get high-resolution detection results. We also develop a token-based multi-task decoder to simultaneously perform saliency and boundary detection by introducing task-related tokens and a novel patch-task-attention mechanism. Experimental results show that our model outperforms existing methods on both RGB and RGB-D SOD benchmark datasets. Most importantly, our whole framework not only provides a new perspective for the SOD field but also shows a new paradigm for transformer-based dense prediction models. Code is available at https://github.com/nnizhang/VST.

1. Introduction

SOD aims to detect objects that attract peoples’ eyes and can help many vision tasks, e.g., [58, 19]. Recently, RGB-D SOD has also gained growing interest with the extra spatial structure information from the depth data. Current state-of-the-art SOD methods are dominated by convolutional architectures [28], on both RGB and RGB-D data. They often adopt an encoder-decoder CNN architecture [47, 57], where the encoder encodes the input image to multi-level features and the decoder integrates the extracted features to predict the final saliency map. Based on this simple architecture, most efforts have been made to build a powerful decoder for predicting better saliency results. To this end, they introduced various attention models [37, 80, 7], multi-scale feature integration methods [24, 49, 16, 43], and multi-task learning frameworks [67, 77, 82, 69, 25]. An additional demand for RGB-D SOD is to effectively fuse cross-modal information, i.e., the appearance information and the depth cues. Existing works propose various modality fusion methods, such as feature fusion [22, 4, 16, 18, 89], knowledge distillation [53], dynamic convolution [48], attention models [31, 78], and graph neural networks [43]. Hence, CNN-based methods have achieved impressive results [66, 88].

However, all previous methods are limited in learning global long-range dependencies. Global contexts [21, 83, 56, 44, 37] and global contrast [75, 2, 8] have been proved crucial for saliency detection for a long time. Nevertheless, due to the intrinsic limitation of CNNs that they extract features in local sliding windows, previous methods can hardly exploit the crucial global cues. Although some methods utilized fully connected layers [36, 22], global pooling layers [44, 37, 65], and non-local modules [38, 7] to incorporate the global context, they only did such in certain layers and the standard CNN-based architecture remains unchanged.

Recently, Transformer [61] was proposed to model global long-range dependencies among word sequences for machine translation. The core idea is the self-attention mechanism, which leverages the query-key correlation to relate different positions in a sequence. Transformer stacks the self-attention layers multiple times in both encoder and decoder, thus can model long-range dependencies in every layer. Hence, it is natural to introduce the Transformer to SOD, leveraging the global cues in the model all the way.

In this paper, for the first time, we rethink SOD from a new sequence-to-sequence perspective and develop a novel unified model for both RGB and RGB-D SOD based on a pure transformer, which is named Visual Saliency Transformer. We follow the recently proposed ViT models [12, 74] to divide each image into patches and adopt the Transformer model on the patch sequence. Then, the Transformer propagates long-range dependencies between image patches, without any need of using convolution. However,
it is not straightforward to apply ViT for SOD. On the one hand, how to perform dense prediction tasks based on pure transformer still remains an open question. On the other hand, ViT usually tokenizes the image to a very coarse scale. How to adapt ViT to the high-resolution prediction demand of SOD is also unclear.

To solve the first problem, we design a token-based transformer decoder by introducing task-related tokens to learn decision embeddings. Then, we propose a novel patch-task-attention mechanism to generate dense-prediction results, which provides a new paradigm for using transformer in dense prediction tasks. Motivated by previous SOD models [82, 87, 79, 25] that leveraged boundary detection to boost the SOD performance, we build a multi-task decoder to simultaneously conduct saliency and boundary detection by introducing a saliency token and a boundary token. This strategy simplifies the multitask prediction workflow by simply learning task-related tokens, thus largely reduces the computational costs while obtaining better results. To solve the second problem, inspired by the Tokens-to-Token (T2T) transformation [74], which reduces the length of tokens, we propose a new reverse T2T transformation to upsample tokens by expanding each token into multiple sub-tokens. Then, we upsample patch tokens progressively and fuse them with low-level tokens to obtain the final full-resolution saliency map. In addition, we also use a cross modality transformer to deeply explore the interaction between multi-modal information for RGB-D SOD. Finally, our VST outperforms existing state-of-the-art SOD methods with a comparable number of parameters and computational costs, on both RGB and RGB-D data.

Our main contributions can be summarized as follows:

- For the first time, we design a novel unified model based on the pure transformer architecture for both RGB and RGB-D SOD, from a new perspective of sequence-to-sequence modeling.
- We design a multi-task transformer decoder to jointly conduct saliency and boundary detection by introducing task-related tokens and patch-task-attention.
- We propose a new token upsampling method for transformer-based framework.
- Our proposed VST model achieves state-of-the-art results on both RGB and RGB-D SOD benchmark datasets, which demonstrates its effectiveness and the potential of transformer-based models for SOD.

2. Related Work

2.1. Deep Learning Based SOD

CNN-based approaches have become a mainstream trend in both RGB and RGB-D SOD and achieved promising performance. Most methods [24, 65, 49, 84, 16] leveraged a multi-level feature fusion strategy by using UNet [57] or HED-style [71] network structures. Some works introduced the attention mechanism to learn more discriminative features, including spatial and channel attention [52, 80, 16, 7] or pixel-wise contextual attention [37]. Other works [36, 64, 11, 42, 6] tried to design recurrent networks to refine the saliency map step-by-step. In addition, some works introduced multi-task learning, e.g., fixation prediction [67], image caption [77], and edge detection [54, 82, 69, 79, 25] to boost the SOD performance.

As for RGB-D SOD, many methods have designed various models to fuse RGB and depth features and obtained significant results. Some models [4, 5, 18] adopted simple feature fusion methods, i.e., concatenation, summation, or multiplication. Some others [81, 30, 52, 31] leveraged the depth cues to generate spatial or channel attention to enhance the RGB features. Besides, dynamic convolution [48], graph neural networks [43], and knowledge distillation [53] were also adopted to implement multi-modal feature fusion. In addition, [38, 39, 7] adopted the cross-attention mechanism to propagate long-range cross-modal interactions between RGB and depth cues.

Different from previous CNN-based methods, we are the first to rethink SOD from a sequence-to-sequence perspective and propose a unified model based on pure transformer for both RGB and RGB-D SOD. In our model, we follow [54, 82, 69, 79, 25] to leverage boundary detection to boost the SOD performance. However, different from these CNN-based models, we design a novel token-based multitask decoder to achieve this goal under the transformer framework.

2.2. Transformers in Computer Vision

Vaswani et al. [61] first proposed a transformer encoder-decoder architecture for machine translation, where multi-head self-attention and point-wise feed-forward layers are stacked multiple times. Recently, more and more works have introduced the Transformer model to various computer vision tasks and achieved excellent results. Some works combined CNNs and transformers into hybrid architectures for object detection [3, 91], panoptic segmentation [62], lane shape prediction [40], and so on. Typically, they first use CNNs to extract image features and then leverage the Transformer to incorporate long-range dependencies.

Other works design pure transformer models to process images from the sequence-to-sequence perspective. ViT [12] divided each image into a sequence of flattened 2D patches and then adopted the Transformer for image classification. Touvron et al. [60] introduced a teacher-student strategy to improve the data-efficiency of ViT and Wang et al. [68] proposed a pyramid architecture to adapt ViT for dense prediction tasks. T2T-ViT [74] adopted the T2T module to model local structures, thus generating multiscale token features. In this work, we adopt T2T-ViT as the backbone and propose a novel multitask decoder and a reverse...
T2T token upsampling method. It is noteworthy that our usage of task-related tokens is different from previous models. In [12, 60], the class token is directly used for image classification via adopting a multilayer perceptron on the token embedding. However, we can not obtain dense prediction results directly from a single task token. Thus, we propose to perform patch-task-attention between patch tokens and the task tokens to predict saliency and boundary maps. We believe our strategy will also inspire future transformer models for other dense prediction tasks.

Another related work to ours is [86], which introduces transformer into the semantic segmentation task. The authors adopted a vision transformer as a backbone and then reshaped the token sequences to 2D image features. Then, they predicted full-resolution segmentation maps using convolution and bilinear upsampling. Their model still falls into the hybrid architecture category. In contrast, our model is a pure transformer architecture and does not rely on any convolution operation and bilinear upsampling.

### 3. Visual Saliency Transformer

Figure 1 shows the overall architecture of our proposed VST model. The main components include a transformer encoder based on T2T-ViT, a transformer convertor to convert patch tokens from the encoder space to the decoder space, and a multi-task transformer decoder.

#### 3.1. Transformer Encoder

Similar to other CNN-based SOD methods, which often utilize pretrained image classification models such as VGG [59] and ResNet [23] as the backbone of their encoders to extract image features, we adopt the pretrained T2T-ViT [74] model as our backbone, as detailed below.

#### 3.1.1 Tokens to Token

Given a sequence of patch tokens $T'$ with length $l$ from the previous layer, T2T-ViT iteratively applies the T2T module, which is composed of a re-structurization step and a soft split step, to model the local structure information in $T'$ and obtain a new sequence of tokens.

**Re-structurization.** As shown in Figure 2(a), the tokens $T'$ is first transformed using a transformer layer to obtain new tokens $T \in \mathbb{R}^{l \times c}$:

$$T = \text{MLP}(\text{MSA}(T')),$$

where MSA and MLP denote the multi-head self-attention and multilayer perceptron in the original Transformer [61], respectively. Note that layer normalization [1] is applied before each block. Then, $T$ is reshaped to a 2D image $I \in \mathbb{R}^{h \times w \times c}$, where $l = h \times w$, to recover spatial structures, as shown in Figure 2(a).

**Soft split.** After the re-structurization step, $I$ is first split into $k \times k$ patches with $s$ overlapping, $p$ zero-padding is also utilized to pad image boundaries. Then, the image patches are unfolded to a sequence of tokens $T_o \in \mathbb{R}^{l_o \times c k^2}$, where the sequence length $l_o$ is computed as:

$$l_o = h_o \times w_o = \left\lfloor\frac{h + 2p - k}{k - s} + 1\right\rfloor \times \left\lfloor\frac{w + 2p - k}{k - s} + 1\right\rfloor.$$  

(2)

Different from ViT [12], the overlapped patch splitting adopted in T2T-ViT introduces local correspondence within neighbouring patches, thus bringing spatial priors.

The T2T transformation can be conducted iteratively multiple times. In each time, the re-structurization step first transforms previous token embeddings to new embeddings
and also integrates long-range dependencies within all tokens. Then, the soft split operation aggregates the tokens in each $k \times k$ neighbour into a new token, which is ready to use for the next layer. Furthermore, when setting $s < k - 1$, the length of tokens can be reduced progressively.

We follow [74] to first soft split the input image into patches and then adopt the T2T module twice. Among the three soft split steps, the patch sizes are set to $k = [7, 3, 3]$, the overlappings are set to $s = [3, 1, 1]$, and the padding sizes are set to $p = [2, 1, 1]$. As such, we can obtain multi-level tokens $T_1 \in \mathbb{R}^{H_1 \times W_1}$, $T_2 \in \mathbb{R}^{H_2 \times W_2}$, and $T_3 \in \mathbb{R}^{H_3 \times W_3}$. Given the width and height of the input image as $H$ and $W$, respectively, then $l_1 = \frac{H}{7} \times \frac{W}{7}$, $l_2 = \frac{H}{3} \times \frac{W}{3}$, and $l_3 = \frac{H}{11} \times \frac{W}{11}$. We follow [74] to set $c = 64$ and use a linear projection layer on $T_3$ to transform its embedding dimension from $c$ to $d = 384$.

### 3.1.2 Encoder with T2T-ViT Backbone

The final token sequence $T_3$ is added with the sinusoidal position embedding [61] to encode 2D position information. Then, $L^C$ transformer layers are used to model long-range dependencies among $T_3$ to extract powerful patch token embeddings $T^c \in \mathbb{R}^{L \times d}$.

For RGB SOD, we adopt a single transformer encoder to obtain RGB encoder patch tokens $T^c_r \in \mathbb{R}^{L \times d}$ from each input RGB image. For RGB-D SOD, we follow two-stream architectures to further use another transformer encoder to extract the depth encoder patch tokens $T^c_d$ from the input depth map in a similar way, as shown in Figure 1.

### 3.2. Transformer Convertor

We insert a convertor module between the transformer encoder and decoder to convert the encoder patch tokens $T^c_r$ from the encoder space to the decoder space, thus obtaining the converted patch tokens $T^c \in \mathbb{R}^{L \times d}$.

#### 3.2.1 RGB-D Convertor

We fuse $T^c_r$ and $T^c_d$ in the RGB-D converter to integrate the complementary information between the RGB and depth data. To this end, we design a Cross Modality Transformer (CMT), which consists of $L^C$ alternating cross-modality-attention layers and self-attention layers.

**Cross-modality-attention.** Under the pure transformer architecture, we modify the standard self-attention layer to propagate long-range cross-modal dependencies between the image and depth data, thus obtaining the cross-modality-attention, which is detailed as follows.

First, similar with the self-attention in [61], $T^c_r$ is embedded to queries $Q_r \in \mathbb{R}^{L \times d}$, keys $K_r \in \mathbb{R}^{L \times d}$, and values $V_r \in \mathbb{R}^{L \times d}$ through three linear projections. Similarly, we can obtain the depth queries $Q_d$, keys $K_d$, and values $V_d$ from $T^c_d$. Next, we compute the “Scaled Dot-Product Attention” [61] between the queries from one modality with the keys from the other modality. Then, the output is computed as a weighted sum of the values, formulated as:

$$
\text{Attention}(Q_r, K_d, V_d) = \text{softmax}(Q_r K_d^T / \sqrt{d}) V_d.
$$

$$
\text{Attention}(Q_d, K_r, V_r) = \text{softmax}(Q_d K_r^T / \sqrt{d}) V_r. \quad (3)
$$

We follow the standard Transformer architecture in [61] and adopt the multi-head attention mechanism in the cross-modality-attention. The same positionwise feed-forward network, residual connections, and layer normalization [1] are also used, forming our CMT layer.

After each adoption of the proposed CMT layer, we use one standard transformer layer on each RGB and depth patch token sequence, further enhancing their token embeddings. After alternately using CMT and transformer for $L^C$ times, we fuse the obtained RGB tokens and depth tokens by concatenation and then project them to the final converted tokens $T^c$, as shown in Figure 1.

#### 3.2.2 RGB Convertor

To align with our RGB-D SOD model, for RGB SOD, we simply use $L^C$ standard transformer layers on $T^c_r$ to obtain the converted patch token sequence $T^c$.

### 3.3. Multi-task Transformer Decoder

Our decoder aims to decode the patch tokens $T^c$ to saliency maps. Hence, we propose a novel token upsam-
pling method with multi-level token fusion and a token-based multi-task decoder.

### 3.3.1 Token Upsampling and Multi-level Token Fusion

We argue that directly predicting saliency maps from $T^c$ can not obtain high-quality results since the length of $T^c$ is relatively small, i.e., $l_3 = \frac{H}{16} \times \frac{W}{16}$, which is limited for dense prediction. Thus, we propose to upsample patch tokens first and then conduct dense prediction. Most CNN-based methods [84, 82, 38, 18] adopt bilinear upsampling to recover large scale feature maps. Alternatively, we propose a new token upsampling method under the transformer framework. Inspired by the T2T module [74] that aggregates neighbour tokens to reduce the length of tokens progressively, we propose a reverse T2T (RT2T) transformation to upsample tokens by expanding each token into multiple sub-tokens, as shown in Figure 2(b).

Specifically, we first project the input patch tokens to reduce their embedding dimension from $d = 384$ to $c = 64$. Then, we use another linear projection to expand the embedding dimension from $c$ to $ck^2$. Next, similar to the soft split step in T2T, each token is seen as a $k \times k$ image patch and neighbouring patches have $s$ overlapping. Then, we can fold the image as input using zero-padding. The output image size can be computed using (2) reversely, can fold the tokens as an image using $s$ and neighbouring patches have image back to the upsampled tokens with size $h \times w \times \mathbb{C}$ given the length of the input patch tokens as $h_o \times w_o$, the spatial size of the out image is $h \times w$. Finally, we reshape the output image size can be computed using (2) reversely, i.e., given the length of the input patch tokens as $h_o \times w_o$, the spatial size of the out image is $h \times w$. By setting $s < k - 1$, the RT2T transformation can increase the length of the tokens. Motivated by T2T-ViT, we use RT2T three times and set $k = [3, 3, 7]$, $s = [1, 1, 3]$, and $p = [1, 1, 3]$. Thus, the length of the patch tokens can be gradually upsampled to $H \times W$, equalling to the original size of the input image.

Furthermore, motivated by the widely proved successes of multi-level feature fusion in existing SOD methods [24, 49, 84, 16, 43], we leverage low-level tokens with larger lengths from the T2T-ViT encoder, i.e., $T_1$ and $T_2$, to provide accurate local structural information. For both RGB and RGB-D SOD, we only use the low-level tokens from the RGB transformer encoder. Concretely, we progressively fuse $T_2$ and $T_1$ with the upsampled patch tokens via concatenation and linear projection. Then, we adopt one transformer layer to obtain the decoder tokens $T^D_i$ at each level $i$, where $i = 2, 1$. The whole process is formulated as:

$$T^D_i = \text{MLP}(\text{MSA}(\text{Linear}([\text{RT2T}(T^D_{i-1}), T_i])), \quad (4)$$

where $[.]$ means concatenation along the token embedding dimension. “Linear” means linear projection to reduce the embedding dimension after the concatenation to $c$. Finally, we use another linear projection to recover the embedding dimension of $T^D_i$ back to $d$.

### 3.3.2 Token Based Multi-task Prediction

Inspired by existing pure transformer methods [74, 12], which add a class token on the patch token sequence for image classification, we also leverage task-related tokens to predict results. However, we can not obtain dense prediction results by directly using MLP on the task token embedding, as done in [74, 12]. Hence, we propose to perform patch-task-attention between the patch tokens and the task-related token to perform SOD.

In addition, motivated by the widely used boundary detection in SOD models [82, 69, 79, 25], we also adopt the multi-task learning strategy to jointly perform saliency and boundary detection, thus using the latter to help boost the performance of the former.

To this end, we design two task-related tokens, i.e., a saliency token $t_s \in \mathbb{R}^{1 \times d}$ and a boundary token $t_b \in \mathbb{R}^{1 \times d}$. At each decoder level $i$, we add the saliency and boundary tokens $t_s$ and $t_b$ to the patch token sequence $T^D_i$, and then process them using $L^D_i$ transformer layers. As such, the two task tokens can learn image-dependent task-related embeddings from the interaction with the patch tokens. After this, we take the updated patch tokens as input and perform the token upsampling and multi-level fusion process in (4) to obtain upsampled patch tokens $T^D_{i-1}$. Next, we reuse the updated $t_s$ and $t_b$ in the next level $i - 1$ to further update them and $T^D_{i-1}$. We repeat this process until we reach the last decoder level with the $\frac{1}{4}$ scale.

For saliency and boundary prediction, we perform patch-task-attention between the final decoder patch tokens $T^D_1$ and the saliency and boundary tokens $t_s$ and $t_b$. For saliency prediction, we first embed $T^D_1$ to queries $Q^D_s \in \mathbb{R}^{1 \times d}$ and embed $t_s$, to a key $K_s \in \mathbb{R}^{1 \times d}$ and a value $V_s \in \mathbb{R}^{1 \times d}$. Similarly, for boundary prediction, we embed $T^D_1$ to $Q^D_b$ and embed $t_b$ to $K_b$ and $V_b$. Then, we adopt the patch-task-attention to obtain the task-related patch tokens:

$$T^D_s = \text{sigmoid}(Q^D_s K_s^T / \sqrt{d}) V_s + T^D_1,$$

$$T^D_b = \text{sigmoid}(Q^D_b K_b^T / \sqrt{d}) V_b + T^D_1. \quad (5)$$

Here we use the sigmoid activation for the attention computation since in each equation we only have one key.

Since $T^D_s$ and $T^D_b$ are at the $\frac{1}{4}$ scale, we adopt the third RT2T transformation to upsample them to the full resolution. Finally, we apply two linear transformations with the sigmoid activation to project them to scalars in $[0, 1]$, and then reshape them to a 2D saliency map and a 2D boundary map, respectively. The whole process is given in Figure 1.

### 4. Experiments

#### 4.1. Datasets and Evaluation Metrics

For RGB SOD, we evaluate our VST model on six widely used benchmark datasets, including ECSSD [72]
Table 1. Ablation studies of our proposed model. “Bili” denotes bilinear upsampling. “F” means multi-level token fusion. “TMD” denotes our proposed token-based multi-task decoder, while “C2D” means using conventional two-stream decoder to perform saliency and boundary detection without using task-related tokens. The best results are labeled in blue.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.869</td>
<td>0.862</td>
<td>0.931</td>
<td>0.073</td>
</tr>
<tr>
<td>+CMT</td>
<td>0.889</td>
<td>0.878</td>
<td>0.942</td>
<td>0.062</td>
</tr>
<tr>
<td>+CMT-Bili</td>
<td>0.926</td>
<td>0.930</td>
<td>0.961</td>
<td>0.032</td>
</tr>
<tr>
<td>+CMT+RT2T</td>
<td>0.934</td>
<td>0.940</td>
<td>0.964</td>
<td>0.028</td>
</tr>
<tr>
<td>+CMT+RT2T+F</td>
<td>0.943</td>
<td>0.948</td>
<td>0.969</td>
<td>0.024</td>
</tr>
<tr>
<td>+CMT+RT2T+F+C2D</td>
<td>0.941</td>
<td>0.947</td>
<td>0.968</td>
<td>0.026</td>
</tr>
</tbody>
</table>

We adopt four widely used evaluation metrics to evaluate our model performance comprehensively. Specifically, Structure-measure $S_m$ [13] evaluates region-aware and object-aware structural similarity. Maximum F-measure (maxF) jointly considers precision and recall and recall under the optimal threshold. Maximum enhanced-alignment measure $E_{\text{max}}$ [14] simultaneously considers pixel-level errors and image-level errors. Mean Absolute Error (MAE) computes pixel-wise average absolute error. To evaluate the model complexity, we also report the multiply accumulate operations (MACs) and the number of parameters (Params).

4.2. Implementation Details

For fair comparisons, we follow most previous methods to use the training set of DUTS to train our VST for RGB SOD and use 1,485 images from NJUD, 700 images from NLPR, and 800 images from DUTLF-Depth to train our VST for RGB-D SOD. We follow [82] to use a sober operator to generate the boundary ground truth from GT saliency maps. For depth data preprocessing, we normalize the depth data to [0, 1] and duplicate them to three channels. Finally, we resize each image or depth map to 256 × 256 pixels and then randomly crop 224 × 224 image regions as the model input and use random flipping as data augmentation.

We use the pre-trained T2T-ViT$_{L}$-14 [74] model as our backbone since it has similar computational complexity as ResNet50 [23] does. This model uses the efficient Performer [10] and $c = 64$ in T2T modules, and sets $L^c = 14$. In our convertor and decoder, we set $L^C = L^P_3 = 4$ and $L^P_2 = L^P_1 = 2$ according to experimental results. We set the batchsizes as 11 and 8, and the total training steps as 40,000 and 60,000, for RGB and RGB-D SOD, respectively. For both of them, Adam [27] is adopted as the optimizer and the binary cross entropy loss is used for both saliency and boundary prediction. The initial learning rate is set to 0.0001 and reduced by a factor of 10 at half and three-quarters of the total step, respectively. Deep supervision is also used to facilitate the model training, where we use the patch-task attention to predict saliency and boundary at each decoder level. We implemented our model using Pytorch [50] and trained it on a GTX 1080 Ti GPU.

4.3. Ablation Study

Since our RGB-D VST is built by adding one more transformer encoder and additional CMT based on our RGB VST, while the other parts of the two models are the same, we conduct ablation studies based on our RGB-D VST to verify all of our proposed model components. The experimental results on four RGB-D SOD datasets, i.e., NJUD, DUTLF-Depth, STERE, and LFSD, are given in Table 1. We remove the transformer convertor and the decoder from our RGB-D VST as the baseline model. Specifically, it uses the two-stream transformer encoder to extract RGB encoder patch tokens $T^c_i$ and the depth encoder patch tokens $T^d_i$, and then directly concatenate them and predict the saliency map with 1/16 scale by using MLP on each patch token.

Effectiveness of CMT. For cross-modal information fusion, we deploy our proposed CMT right after the transformer encoder to substitute the concatenation fusion method in the baseline model, shown as “+CMT” in Table 1. Compared to the baseline, CMT brings performance gain especially on the NJUD and LFSD datasets, hence demonstrating its effectiveness.

Effectiveness of RT2T. Based on “+CMT” model, we further simply use bilinear upsampling (“+CMT+Bili”) to progressively upsample tokens to the full resolution and then predict the saliency map. The results show using bilinear upsampling to increase the resolution of the saliency map can largely improve the model performance. Then, we replace bilinear upsampling with our proposed RT2T token upsampling method (“+CMT+RT2T”). We find that RT2T leads to obvious performance improvement compared with using bilinear upsampling, which verifies its effectiveness.

Effectiveness of multi-level token fusion. We progressively fuse $T_1$ and $T_2$ in our decoder (“+CMT+RT2T+F”) to...
Table 2. Quantitative comparison of our proposed VST with other 14 SOTA RGB-D SOD methods on 9 benchmark datasets. Red and blue denote the best and the second-best results, respectively. '-' indicates the code or result is not available.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>A2dele JL-DCF SSF-RGBD UC-Net 5th MA PGAR DANet cmMS AISty CMW Cas-Gnn HDINet CoNet BBS-Net</th>
<th>MACs (G)</th>
<th>Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VST</td>
<td></td>
<td></td>
<td>41.86</td>
<td>143.52</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>A2dele JL-DCF SSF-RGBD UC-Net 5th MA PGAR DANet cmMS AISty CMW Cas-Gnn HDINet CoNet BBS-Net</th>
<th>MACs (G)</th>
<th>Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VST</td>
<td></td>
<td></td>
<td>30.34</td>
<td>132.93</td>
</tr>
</tbody>
</table>

Supply low-level fine-grained information. We find that this strategy further improves the model performance. Hence, leveraging low-level tokens in transformer is as important as fusing low-level features in CNN-based models.

**Effectiveness of the multi-task transformer decoder.**

Based on "$\text{"+CMT+RT2T+F"}$, we further use our token-based multi-task decoder (TMD) to jointly perform saliency and boundary detection ("\$\text{"+CMT+RT2T+F+TMD"}\$"). It shows that using boundary detection can bring further performance gain for SOD on three out of four datasets. To verify the effectiveness of our token-based prediction scheme, we try to directly use a conventional two-stream decoder (C2D) by using the "$\text{"+RT2T+F"}$ architecture twice to predict the saliency map and boundary map via MLP, without using task-related tokens. This model is denoted as "$\text{"+CMT+RT2T+F+C2D"}$ in Table 1. The parameters and MACs of TMD vs. C2D are 17.22 M vs. 20.35 M and 17.70 G vs. 28.27 G, respectively. The results show that using our TMD can achieve better results than using C2D on three out of four datasets, and also with much less computational costs. This clearly demonstrates the superiority of our proposed token-based transformer decoder.

### 4.4 Comparison with State-of-the-Art Methods

For RGB-D SOD, we compare our VST with 14 state-of-the-art RGB-D SOD methods, i.e., A2dele [53], JL-DCF [18], SSF-RGBD [79], UC-Net [76], 5th MA [38], PGAR [6], DANet [85], cmMS [29], AISty [78], CMW [31], Cas-Gnn [43], HDINet [48], CoNet [25], and BBS-Net [16]. For RGB SOD, we compare our VST with 12 state-of-the-art RGB SOD models, including GateNet [84], CSF [20], LDF [69], MINet [49], ITSD [87], EGNet [82], TSPOANet [41], AFNet [17], PoolNet [35], CPD [70], BASNet [55], and PicANet [37]. Table 2 and Table 3 show the quantitative comparison results for RGB-D and RGB SOD, respectively.
Table 3. Quantitative comparison of our proposed VST with other 12 SOTA RGB SOD methods on 6 benchmark datasets. “-R” and “-R2” means the ResNet50 and Res2Net backbone, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>FICANet</th>
<th>BASNet</th>
<th>CPD-R</th>
<th>PoolNet</th>
<th>AFNet</th>
<th>TSPOANet</th>
<th>EGNet-R</th>
<th>ITSD-R</th>
<th>MINet-R</th>
<th>LDF-R</th>
<th>CSF-R2</th>
<th>GateNet-R</th>
<th>VST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MACs (G)</td>
<td>54.05</td>
<td>127.36</td>
<td>17.77</td>
<td>88.89</td>
<td>21.66</td>
<td>-</td>
<td>157.21</td>
<td>15.96</td>
<td>87.11</td>
<td>15.51</td>
<td>19.86</td>
<td>162.13</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Params (M)</td>
<td>47.22</td>
<td>87.06</td>
<td>47.85</td>
<td>68.26</td>
<td>35.95</td>
<td>-</td>
<td>111.64</td>
<td>26.47</td>
<td>162.38</td>
<td>25.15</td>
<td>36.53</td>
<td>128.63</td>
<td>-</td>
</tr>
<tr>
<td>DUTS</td>
<td>$S_m \uparrow$</td>
<td>0.863</td>
<td>0.869</td>
<td>0.869</td>
<td>0.879</td>
<td>0.879</td>
<td>0.860</td>
<td>0.887</td>
<td>0.885</td>
<td>0.884</td>
<td>0.892</td>
<td>0.890</td>
<td>0.891</td>
<td>0.896</td>
</tr>
<tr>
<td></td>
<td>maxF$^\xi$</td>
<td>0.840</td>
<td>0.838</td>
<td>0.840</td>
<td>0.853</td>
<td>0.838</td>
<td>0.828</td>
<td>0.866</td>
<td>0.867</td>
<td>0.864</td>
<td>0.877</td>
<td>0.869</td>
<td>0.874</td>
<td>0.877</td>
</tr>
<tr>
<td></td>
<td>$E_m \uparrow$</td>
<td>0.915</td>
<td>0.902</td>
<td>0.913</td>
<td>0.917</td>
<td>0.910</td>
<td>0.907</td>
<td>0.926</td>
<td>0.929</td>
<td>0.926</td>
<td>0.930</td>
<td>0.929</td>
<td>0.932</td>
<td>0.939</td>
</tr>
<tr>
<td>[63]</td>
<td>MAEL</td>
<td>0.040</td>
<td>0.047</td>
<td>0.043</td>
<td>0.041</td>
<td>0.045</td>
<td>0.049</td>
<td>0.039</td>
<td>0.041</td>
<td>0.037</td>
<td>0.034</td>
<td>0.037</td>
<td>0.038</td>
<td>0.037</td>
</tr>
<tr>
<td>ECSSD</td>
<td>$S_m \uparrow$</td>
<td>0.873</td>
<td>0.916</td>
<td>0.916</td>
<td>0.918</td>
<td>0.917</td>
<td>0.914</td>
<td>0.907</td>
<td>0.925</td>
<td>0.925</td>
<td>0.925</td>
<td>0.931</td>
<td>0.924</td>
<td>0.932</td>
</tr>
<tr>
<td></td>
<td>maxF$^\xi$</td>
<td>0.929</td>
<td>0.931</td>
<td>0.926</td>
<td>0.929</td>
<td>0.924</td>
<td>0.921</td>
<td>0.936</td>
<td>0.939</td>
<td>0.938</td>
<td>0.942</td>
<td>0.935</td>
<td>0.944</td>
<td>0.944</td>
</tr>
<tr>
<td>[72]</td>
<td>MAEL</td>
<td>0.035</td>
<td>0.037</td>
<td>0.037</td>
<td>0.042</td>
<td>0.042</td>
<td>0.047</td>
<td>0.037</td>
<td>0.035</td>
<td>0.034</td>
<td>0.034</td>
<td>0.033</td>
<td>0.038</td>
<td>0.034</td>
</tr>
<tr>
<td>HKU-IS</td>
<td>$S_m \uparrow$</td>
<td>0.913</td>
<td>0.919</td>
<td>0.919</td>
<td>0.911</td>
<td>0.920</td>
<td>0.910</td>
<td>0.909</td>
<td>0.923</td>
<td>0.926</td>
<td>0.926</td>
<td>0.929</td>
<td>-</td>
<td>0.937</td>
</tr>
<tr>
<td></td>
<td>maxF$^\xi$</td>
<td>0.951</td>
<td>0.952</td>
<td>0.950</td>
<td>0.955</td>
<td>0.949</td>
<td>0.945</td>
<td>0.956</td>
<td>0.960</td>
<td>0.960</td>
<td>0.958</td>
<td>-</td>
<td>0.959</td>
<td>0.968</td>
</tr>
<tr>
<td>[32]</td>
<td>MAEL</td>
<td>0.031</td>
<td>0.032</td>
<td>0.034</td>
<td>0.032</td>
<td>0.036</td>
<td>0.039</td>
<td>0.031</td>
<td>0.031</td>
<td>0.029</td>
<td>0.028</td>
<td>-</td>
<td>0.031</td>
<td>0.030</td>
</tr>
<tr>
<td>PASCAL-S</td>
<td>$S_m \uparrow$</td>
<td>0.846</td>
<td>0.837</td>
<td>0.847</td>
<td>0.852</td>
<td>0.849</td>
<td>0.841</td>
<td>0.852</td>
<td>0.861</td>
<td>0.865</td>
<td>0.861</td>
<td>0.863</td>
<td>0.863</td>
<td>0.873</td>
</tr>
<tr>
<td></td>
<td>maxF$^\xi$</td>
<td>0.824</td>
<td>0.819</td>
<td>0.817</td>
<td>0.830</td>
<td>0.824</td>
<td>0.817</td>
<td>0.825</td>
<td>0.839</td>
<td>0.831</td>
<td>0.839</td>
<td>0.839</td>
<td>0.839</td>
<td>0.836</td>
</tr>
<tr>
<td></td>
<td>MAEL</td>
<td>0.072</td>
<td>0.083</td>
<td>0.077</td>
<td>0.076</td>
<td>0.076</td>
<td>0.082</td>
<td>0.080</td>
<td>0.071</td>
<td>0.071</td>
<td>0.067</td>
<td>0.073</td>
<td>0.071</td>
<td>0.067</td>
</tr>
<tr>
<td>DUT-O</td>
<td>$S_m \uparrow$</td>
<td>0.826</td>
<td>0.836</td>
<td>0.825</td>
<td>0.832</td>
<td>0.826</td>
<td>0.818</td>
<td>0.841</td>
<td>0.840</td>
<td>0.833</td>
<td>0.839</td>
<td>0.838</td>
<td>0.840</td>
<td>0.850</td>
</tr>
<tr>
<td></td>
<td>maxF$^\xi$</td>
<td>0.767</td>
<td>0.779</td>
<td>0.754</td>
<td>0.769</td>
<td>0.759</td>
<td>0.750</td>
<td>0.778</td>
<td>0.792</td>
<td>0.769</td>
<td>0.782</td>
<td>0.775</td>
<td>0.782</td>
<td>0.800</td>
</tr>
<tr>
<td>[73]</td>
<td>MAEL</td>
<td>0.054</td>
<td>0.057</td>
<td>0.056</td>
<td>0.056</td>
<td>0.057</td>
<td>0.062</td>
<td>0.053</td>
<td>0.061</td>
<td>0.056</td>
<td>0.052</td>
<td>0.055</td>
<td>0.055</td>
<td>0.058</td>
</tr>
<tr>
<td>SOD</td>
<td>$S_m \uparrow$</td>
<td>0.813</td>
<td>0.799</td>
<td>0.797</td>
<td>0.823</td>
<td>0.811</td>
<td>0.802</td>
<td>0.824</td>
<td>0.835</td>
<td>0.830</td>
<td>0.831</td>
<td>0.826</td>
<td>0.827</td>
<td>0.834</td>
</tr>
<tr>
<td></td>
<td>maxF$^\xi$</td>
<td>0.838</td>
<td>0.808</td>
<td>0.834</td>
<td>0.839</td>
<td>0.838</td>
<td>0.835</td>
<td>0.831</td>
<td>0.849</td>
<td>0.835</td>
<td>0.831</td>
<td>0.832</td>
<td>0.835</td>
<td>0.866</td>
</tr>
<tr>
<td>[45]</td>
<td>MAEL</td>
<td>0.073</td>
<td>0.091</td>
<td>0.089</td>
<td>0.085</td>
<td>0.085</td>
<td>0.094</td>
<td>0.080</td>
<td>0.075</td>
<td>0.074</td>
<td>0.071</td>
<td>0.079</td>
<td>0.079</td>
<td>0.065</td>
</tr>
</tbody>
</table>

Figure 3. Qualitative comparison against state-of-the-art RGB-D (left) and RGB (right) SOD methods. (GT: ground truth)

The results show that our VST outperforms all previous state-of-the-art CNN-based SOD models on both RGB and RGB-D benchmark datasets, with comparable number of parameters and relatively small MACs, hence demonstrating the great effectiveness of our VST. We also show visual comparison results among best-performed models in Figure 3. It shows our proposed VST can accurately detect salient objects in very challenging scenarios, e.g., big salient objects, cluttered backgrounds, foreground and background having similar appearances, etc.

5. Conclusion

In this paper, we are the first to rethink SOD from a sequence-to-sequence perspective and develop a novel unified model based on a pure transformer, for both RGB and RGB-D SOD. To handle the difficulty of applying transformers in dense prediction tasks, we propose a new token upsampling method under the transformer framework and fuse multi-level patch tokens. We also design a multi-task decoder by introducing task-related tokens and a novel patch-task-attention mechanism to jointly perform saliency and boundary detection. Our VST model achieves state-of-the-art results for both RGB and RGB-D SOD without relying on heavy computational costs, thus showing its great effectiveness. We also set a new paradigm for the open question of how to use transformer in dense prediction tasks.

Acknowledgments: This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0105702, the National Science Foundation of China under Grant 62027813, 62036005, U20B2065, U20B2068.
References


