From Layers to States: A State Space Model Perspective to Deep Neural Network Layer Dynamics

Qinshuo Liu

Department of Statistics and Actuarial Science, HKU

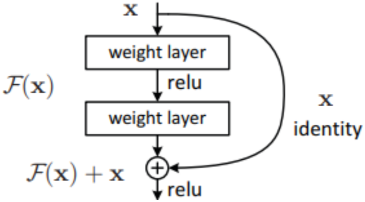
April 2, 2025

Supervised by Prof. Guodong Li

Outline

A Brief History of Layer Interaction

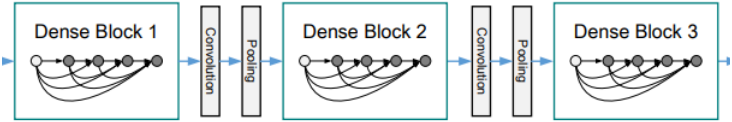
- The growing evidence indicates that strengthening layer interactions can encourage the information flow of a deep neural network.
  - ResNet[1]: employed skip connections, allowing gradients to flow more easily by connecting non-adjacent layers.



[1] Deep residual learning for image recognition..

A Brief History of Layer Interaction

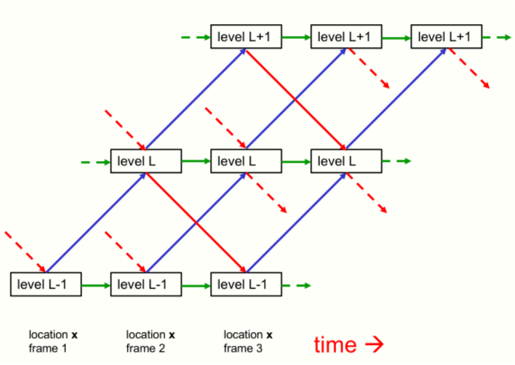
- The growing evidence indicates that strengthening layer interactions can encourage the information flow of a deep neural network.
  - DenseNet[2]: extended this concept further by enabling each layer to access all preceding layers within a stage, fostering a rich exchange of information.



[2] **Densely Connected Convolutional Networks**.

A Brief History of Layer Interaction

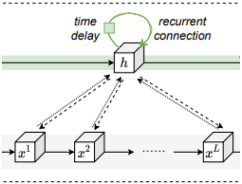
- The growing evidence indicates that strengthening layer interactions can encourage the information flow of a deep neural network.
  - GLOM[3]: proposed an intensely interactive architecture that incorporates bottom-up, top-down, and same-level connections to effectively represent part-whole hierarchies.

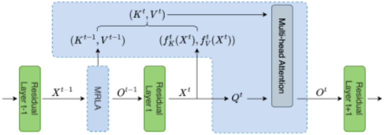


[3] How to represent part-whole hierarchies in a neural network.

A Brief History of Layer Interaction

- The growing evidence indicates that strengthening layer interactions can encourage the information flow of a deep neural network.
  - RLA[4] and MRLA[5]: some studies have begun to frame layer interactions with recurrent models and attention mechanisms.





[4] Recurrence along depth: Deep convolutional neural networks with recurrent layer aggregation.

[5] Cross-Layer Retrospective Retrieving via Layer Attention.

Contributions

- For a deep neural network, we treat the outputs from layers as states of a continuous process and attempt to leverage the SSM to design the aggregation of layers. To our best knowledge, this is the first time such a perspective has been presented.
- This leads to a proposed lightweight module, the Selective State Space Model Layer Aggregation (S6LA) module, and it conceptualizes a neural network as a selective state space model (S6), hence solving the layer interactions by the long sequence modelling selective mechanism.
- Compared with other SOTA convolutional and transformer-based layer aggregation models, S6LA demonstrates superior performance in classification, detection, and instance segmentation tasks.

Revisiting State Space Model

State space model defines:

$$h'(t) = Ah(t) + Bx(t), \tag{1}$$

Revisiting State Space Model

State space model defines:

$$h'(t) = Ah(t) + Bx(t), \tag{1}$$

Discretization:

$$h^t = e^{\Delta A} h^{t-1} + \int_{t-1}^{t} e^{A(t-\tau)} Bx(\tau) d\tau. \tag{2}$$

Revisiting State Space Model

State space model defines:

$$h'(t) = Ah(t) + Bx(t), \tag{1}$$

Discretization:

$$h^t = e^{\Delta A} h^{t-1} + \int_{t-1}^{t} e^{A(t-\tau)} Bx(\tau) d\tau. \tag{2}$$

With zero-order hold (ZOH) condition:

$$h^t = e^{\Delta A} h^{t-1} + (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B x^t. \tag{3}$$

Revisiting State Space Model

State space model defines:

$$h'(t) = Ah(t) + Bx(t), \tag{1}$$

Discretization:

$$h^t = e^{\Delta A} h^{t-1} + \int_{t-1}^{t} e^{A(t-\tau)} Bx(\tau) d\tau. \tag{2}$$

With zero-order hold (ZOH) condition:

$$h^t = e^{\Delta A} h^{t-1} + (\Delta A)^{-1} (\exp(\Delta A) - I) \cdot \Delta B x^t. \tag{3}$$

Then:

$$h^t = \overline{A} h^{t-1} + \overline{B} x^t \tag{4}$$

Revisiting State Space Model

State space model defines:

$$h'(t) = Ah(t) + Bx(t), \tag{1}$$

Discretization:

$$h^t = e^{\Delta A} h^{t-1} + \int_{t-1}^{t} e^{A(t-\tau)} Bx(\tau) d\tau. \tag{2}$$

With zero-order hold (ZOH) condition:

$$h^t = e^{\Delta A} h^{t-1} + (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B x^t. \tag{3}$$

Then:

$$h^t = \overline{A} h^{t-1} + \overline{B} x^t \tag{4}$$

Using the first-order Taylor series:

$$\overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \approx (\Delta A)^{-1}(\Delta A) \cdot \Delta B = \Delta B. \tag{5}$$

CNN Layer Aggregation

The layer aggregation at the tth layer below:

$$A^t = g^t(\mathbf{X}^0, \mathbf{X}^1, \cdots, \mathbf{X}^{t-2}, \mathbf{X}^{t-1}), \quad \mathbf{X}^t = f^t(A^{t-1}, \mathbf{X}^{t-1}), \tag{6}$$

where g^t is used to summarize the first t layers, A^t is the aggregated information, and f^t produces the new layer output from the last hidden layer and the given aggregation which contains the previous information.

CNN Layer Aggregation Example

DenseNet is the first one for the layer aggregation, and its output at tth layer can be formulated into:

$$\mathbf{X}^t = \text{Conv3}^t[\text{Conv1}^t(\text{Concat}(\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^{t-1}))]. \tag{7}$$

Let $A^t = \text{Conv1}^t(\text{Concat}(\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^{t-1}))$ and $\mathbf{X}^t = \text{Conv3}^t(A^t)$.

CNN Layer Aggregation Example

DenseNet is the first one for the layer aggregation, and its output at tth layer can be formulated into:

$$\mathbf{X}^t = \text{Conv3}^t[\text{Conv1}^t(\text{Concat}(\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^{t-1}))]. \tag{7}$$

Let $A^t = \text{Conv1}^t(\text{Concat}(\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^{t-1}))$ and $\mathbf{X}^t = \text{Conv3}^t(A^t)$. Then under the condition if $A^t = \sum_{i=0}^{t-1} \text{Conv1}_i^{t+1}(\mathbf{X}^i)$, a lightweight form is:

$$\mathbf{X}^t = \text{Conv3}^t[A^{t-1} + \text{Conv1}_{t-1}^t(\mathbf{X}^{t-1})]. \tag{8}$$

CNN Layer Aggregation Example

DenseNet is the first one for the layer aggregation, and its output at tth layer can be formulated into:

$$\mathbf{X}^t = \text{Conv3}^t[\text{Conv1}^t(\text{Concat}(\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^{t-1}))]. \tag{7}$$

Let $A^t = \text{Conv1}^t(\text{Concat}(\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^{t-1}))$ and $\mathbf{X}^t = \text{Conv3}^t(A^t)$. Then under the condition if $A^t = \sum_{i=0}^{t-1} \text{Conv1}_i^{t+1}(\mathbf{X}^i)$, a lightweight form is:

$$\mathbf{X}^t = \text{Conv3}^t[A^{t-1} + \text{Conv1}_{t-1}^t(\mathbf{X}^{t-1})]. \tag{8}$$

Therefore, we can treat the update of $\mathbf{X}^t = \mathbf{X}^{t-1} + f^{t-1}(\mathbf{X}^{t-1})$ with applying the update recursively as $A^t = \sum_{i=0}^{t-1} f^i(\mathbf{X}^i) + \mathbf{X}^0$ and $\mathbf{X}^t = A^{t-1} + \mathbf{X}^{t-1}$.

Attention Layers Aggregation

Consider a simple attention layer: $\mathbf{X} \in \mathbb{R}^{L \times D}$, $\mathbf{O} \in \mathbb{R}^{L \times D}$. The output $\mathbf{O}$ has the following mathematical formulation:

$$\mathbf{O} = \text{Self-Attention}(\mathbf{X}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}})\mathbf{V}. \tag{9}$$

Attention Layers Aggregation

Consider a simple attention layer: $\mathbf{X} \in \mathbb{R}^{L \times D}$, $\mathbf{O} \in \mathbb{R}^{L \times D}$. The output $\mathbf{O}$ has the following mathematical formulation:

$$\mathbf{O} = \text{Self-Attention}(\mathbf{X}) = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{D}})\mathbf{V}. \tag{9}$$

A vanilla transformer can then be formulated into:

$$A^t = \mathbf{X}^{t-1} + \text{Self-Attention}(\mathbf{X}^{t-1}), \quad \mathbf{X}^t = A^t + \text{MLP}(\text{Norm}(A^t)). \tag{10}$$

Attention Layers Aggregation

Consider a simple attention layer: $\mathbf{X} \in \mathbb{R}^{L \times D}$, $\mathbf{O} \in \mathbb{R}^{L \times D}$. The output $\mathbf{O}$ has the following mathematical formulation:

$$\mathbf{O} = \text{Self-Attention}(\mathbf{X}) = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{D}})\mathbf{V}. \tag{9}$$

A vanilla transformer can then be formulated into:

$$A^t = \mathbf{X}^{t-1} + \text{Self-Attention}(\mathbf{X}^{t-1}), \quad \mathbf{X}^t = A^t + \text{MLP}(\text{Norm}(A^t)). \tag{10}$$

$$\mathbf{X}^t = f^t(g^t(\mathbf{X}^0, \cdots, \mathbf{X}^{t-1})), \tag{11}$$

where

g^t: the attention layer;

f^t: Add & Norm layer for the t-th layer.

Attention Layers Aggregation

Consider a simple attention layer: $\mathbf{X} \in \mathbb{R}^{L \times D}$, $\mathbf{O} \in \mathbb{R}^{L \times D}$. The output $\mathbf{O}$ has the following mathematical formulation:

$$\mathbf{O} = \text{Self-Attention}(\mathbf{X}) = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{D}})\mathbf{V}. \tag{9}$$

A vanilla transformer can then be formulated into:

$$A^t = \mathbf{X}^{t-1} + \text{Self-Attention}(\mathbf{X}^{t-1}), \quad \mathbf{X}^t = A^t + \text{MLP}(\text{Norm}(A^t)). \tag{10}$$

$$\mathbf{X}^t = f^t(g^t(\mathbf{X}^0, \cdots, \mathbf{X}^{t-1})), \tag{11}$$

where

g^t: the attention layer;

f^t: Add & Norm layer for the t-th layer.

Then $A^t = g^t(\mathbf{X}^0, \cdots, \mathbf{X}^{t-1})$ by $A^t = A^{t-1} + g^{t-1}(\mathbf{X}^{t-1})$.

The Formula of S6LA

we propose our selective state space model layer aggregation below:

$$h^t = g^t(h^{t-1}, \mathbf{X}^t), \quad \mathbf{X}^t = f^t(h^{t-1}, \mathbf{X}^{t-1}), \tag{12}$$

where h^t is a hidden state similar to A^t, g^t is the relation function between the current SSM hidden layer state and previous hidden layer state with input.

The Formula of S6LA

we propose our selective state space model layer aggregation below:

$$h^t = g^t(h^{t-1}, \mathbf{X}^t), \quad \mathbf{X}^t = f^t(h^{t-1}, \mathbf{X}^{t-1}), \tag{12}$$

where h^t is a hidden state similar to A^t, g^t is the relation function between the current SSM hidden layer state and previous hidden layer state with input.

Then the update of h^t can be formulated as:

$$h^t = \overline{A}h^{t-1} + \overline{B}\mathbf{X}^t, \quad \mathbf{X}^t = f^t(h^{t-1}, \mathbf{X}^{t-1}). \tag{13}$$

Overview

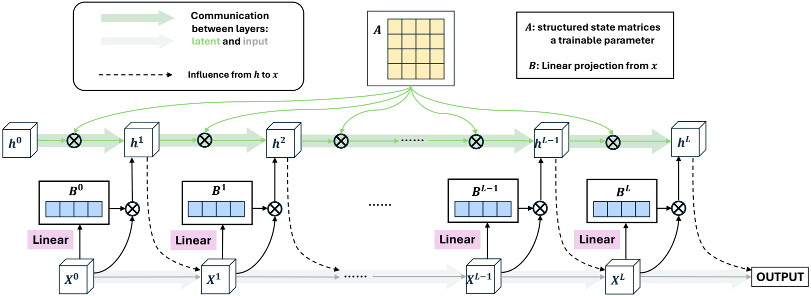


Figure: Schematic diagram of a Network with Selective State Space Model Layer Aggregation.

Leveraging S6LA with CNNs Backbones

Given $\mathbf{X}^t \in \mathbb{R}^{H\times W\times D}$ and $\mathbf{X}^t$ with the state $h^{t-1} \in \mathbb{R}^{H\times W\times N}$ from the previous layer:

- H and W: the height and width;
- D: the embedding dimension;
- N: the dimension of latent states.

Leveraging S6LA with CNNs Backbones

- **Input Treatment:** Merging the input $\mathbf{X}^t$ and the hidden state h^{t-1} through a simple concatenation along the feature dimension to $\mathbf{O}^t$.

Leveraging S6LA with CNNs Backbones

- **Input Treatment:** Merging the input $\mathbf{X}^t$ and the hidden state h^{t-1} through a simple concatenation along the feature dimension to $\mathbf{O}^t$.
- **Latent State Update:** Define $\mathbf{X}^{t+1}$ as the sum of $\mathbf{X}^t$ and $\mathbf{O}^t$.

$$h^t = e^{(\Delta A)} h^{t-1} + \Delta B \mathbf{O}^t, \tag{14}$$

  where $\Delta = W_\Delta(\mathrm{Conv}(\mathbf{O}^t)), B = W_B(\mathrm{Conv}(\mathbf{O}^t))$.
- **Output Computation:** $\mathbf{X}^{t+1} = \mathbf{O}^t + \mathbf{X}^t$.

S6LA with CNNs Backbone

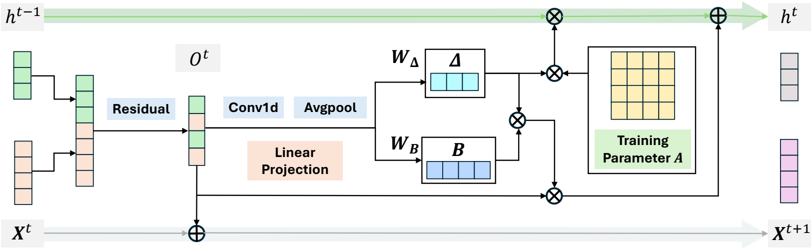


Figure: Detailed operations in S6LA module with Convolutional Neural Network. The green arrow shows the hidden state connection, while the grey arrow indicates layers communications.

Leveraging S6LA with Deep ViTs

- **Input Treatment:**

$$\begin{aligned} \mathbf{X}^t_{\text{input}} &= \text{Add\&Norm}(\text{MLP}(\text{Add\&Norm}(\text{Attn}(\mathbf{X}^t)))); \\ \mathbf{X}^t_p, \mathbf{X}^t_c &= \text{Split}(\mathbf{X}^t_{\text{input}}). \end{aligned} \tag{15}$$

Leveraging S6LA with Deep ViTs

- **Input Treatment:**

$$\begin{aligned} \mathbf{X}^t_{\text{input}} &= \text{Add\&Norm}(\text{MLP}(\text{Add\&Norm}(\text{Attn}(\mathbf{X}^t)))); \\ \mathbf{X}^t_p, \mathbf{X}^t_c &= \text{Split}(\mathbf{X}^t_{\text{input}}). \end{aligned} \tag{15}$$

- **Latent State Update:**

$$h^t = e^{(\Delta A)} h^{t-1} + \Delta B \mathbf{X}^t_c, \tag{16}$$

where: $\Delta = W_\Delta(\mathbf{X}^t_c), B = W_B(\mathbf{X}^t_c)$.

Leveraging S6LA with Deep ViTs

- **Input Treatment:**

$$\begin{aligned} \mathbf{X}^t_{\text{input}} &= \text{Add\&Norm}(\text{MLP}(\text{Add\&Norm}(\text{Attn}(\mathbf{X}^t))));\\ \mathbf{X}^t_p, \mathbf{X}^t_c &= \text{Split}(\mathbf{X}^t_{\text{input}}). \end{aligned} \tag{15}$$

- **Latent State Update:**

$$h^t = e^{(\Delta A)} h^{t-1} + \Delta B \mathbf{X}^t_c, \tag{16}$$

where: $\Delta = W_\Delta(\mathbf{X}^t_c), B = W_B(\mathbf{X}^t_c)$.

- **Output Computation:**

$$\widehat{\mathbf{X}}^t_p = \mathbf{X}^t_p + W\mathbf{X}^t_p h^t. \tag{17}$$

Then,

$$\mathbf{X}^{t+1} = \text{Concat}(\widehat{\mathbf{X}}^t_p, \mathbf{X}^t_c). \tag{18}$$

S6LA with Deep ViTs Backbones

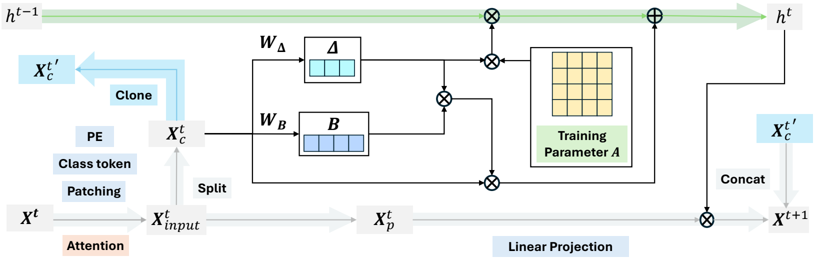


Figure: Diagram of the S6LA architecture with Transformer.

Backbone and Settings

Backbone:

- CNN backbone:
  - ResNet.
- Transformer-based backbone:
  - DeiT;
  - Swin Transformer;
  - PVTv2.

Settings:

N: the dimension of h: $N = 16, 32, 64$ for ResNet, we choose 32 as our baseline feature channel.

Results

Table: Comparisons of the Top-1 and Top-5 accuracy on the ImageNet-1K validation set with CNNs.

Model	Method	Params	FLOPs	Top-1 Acc.	Top-5 Acc.
ResNet-50	Vanilla	25.6 M	4.1 B	76.1	92.9
	+ SE	28.1 M	4.1 B	76.7	93.4
	+ CBAM	28.1 M	4.1 B	77.3	93.7
	+ A^2	34.6 M	7.0 B	77.0	93.5
	+ AA	27.1 M	4.5 B	77.4	93.6
	+ 1 NL	29.0 M	4.4 B	77.2	93.5
	+ 1 GC	26.9 M	4.1 B	77.3	93.5
	+ all GC	29.4 M	4.2 B	77.7	93.7
	+ ECA	25.6 M	4.1 B	77.4	93.6
	+ RLA	25.9 M	4.3 B	77.2	93.4
	+ MRLA	25.7 M	4.6 B	77.5	93.7
	+ S6LA (Ours)	25.8 M	4.4 B	**78.0**	**94.2**
ResNet-101	Vanilla	44.5 M	7.8 B	77.4	93.5
	+ SE	49.3 M	7.8 B	77.6	93.9
	+ CBAM	49.3 M	7.9 B	78.5	94.3
	+ AA	47.6 M	8.6 B	78.7	94.4
	+ ECA	44.5 M	7.8 B	78.7	94.3
	+ RLA	45.0 M	8.2 B	78.5	94.2
	+ MRLA	44.9 M	8.5 B	78.7	94.4
	+ S6LA (Ours)	45.0 M	8.3 B	**79.1**	**94.8**
ResNet-152	Vanilla	60.2 M	11.6 B	78.3	94.0
	+ SE	66.8 M	11.6 B	78.4	94.3
	+ CBAM	66.8 M	11.6 B	78.8	94.4
	+ AA	66.6 M	11.9 B	79.0	94.6
	+ ECA	60.2 M	11.6 B	78.9	94.5
	+ RLA	60.8 M	12.1 B	78.8	94.4
	+ MRLA	60.7 M	12.4 B	79.1	94.6
	+ S6LA (Ours)	60.8 M	12.2 B	**79.4**	**94.9**

Results

Table: Comparisons of the Top-1 and Top-5 accuracy on the ImageNet-1K validation set with vision transformer-based models.

Backbone	Method	Params	FLOPs	Top-1	Top-5
DeiT	DeiT-Ti	5.7 M	1.2 B	72.6	91.1
	+ MRLA	5.7 M	1.4 B	73.0	91.7
	+ S6LA (Ours)	6.1 M	1.5 B	**73.3**	**92.0**
	DeiT-S	22.1 M	4.5 B	79.9	95.0
	+ MRLA	22.1 M	4.6 B	80.7	95.3
	+ S6LA (Ours)	23.3 M	4.8 B	**81.3**	**96.0**
	DeiT-B	86.4 M	16.8 B	81.8	95.6
	+ MRLA	86.5 M	16.9 B	82.9	96.3
	+ S6LA (Ours)	86.9 M	17.1 B	**83.3**	**96.5**
Swin	Swin-T	28.3 M	4.5 B	81.0	95.4
	+ MRLA	28.9 M	4.5 B	80.9	95.2
	+ S6LA (Ours)	30.5 M	4.5 B	**81.5**	**95.6**
	Swin-S	49.6 M	8.7 B	82.8	96.1
	+ MRLA	50.9 M	8.7 B	82.5	96.0
	+ S6LA (Ours)	52.5 M	8.7 B	**83.3**	**96.5**
	Swin-B	87.8 M	15.4 B	83.2	96.4
	+ MRLA	89.8 M	15.5 B	82.9	96.3
	+ S6LA (Ours)	91.3 M	15.5 B	**83.5**	**96.6**
PVTv2	PVTv2-B0	3.4 M	0.6 B	70.0	89.7
	+ MRLA	3.4 M	0.9 B	70.6	90.0
	+ S6LA (Ours)	3.8 M	0.6 B	**70.8**	**90.2**
	PVTv2-B1	13.1 M	2.3 B	78.3	94.3
	+ MRLA	13.2 M	2.4 B	**78.9**	**94.9**
	+ S6LA (Ours)	14.5 M	2.2 B	78.8	94.6
	PVTv2-B2	25.4 M	4.0 B	81.4	95.5
	+ MRLA	25.5 M	4.2 B	81.6	95.2
	+ S6LA (Ours)	26.1 M	4.1 B	**82.3**	**95.9**

Backbone and Settings

Backbone:

- This subsection validates the transferability and the generalization ability of our model on object detection and segmentation tasks using the three typical detection frameworks: Faster R-CNN, RetinaNet and Mask R-CNN.
- Toolkits: MMDetection.

Results

Method	Detector	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
ResNet-50 (He et al., 2016a)	Faster R-CNN	36.4	58.2	39.2	21.8	40.0	46.2
+ SE (Hu et al., 2018)		37.7	59.1	40.9	22.9	41.9	48.2
+ ECA (Wang et al., 2020b)		38.0	60.6	40.9	23.4	42.1	48.0
+ RLA (Zhao et al., 2021)		38.8	59.6	42.0	22.5	42.9	49.5
+ MRLA (Fang et al., 2023)		40.1	61.3	**43.8**	24.0	43.9	52.2
+ S6LA (Ours)		**40.3**	**61.7**	**43.8**	**24.2**	**44.0**	**52.5**
ResNet-101 (He et al., 2016a)		38.7	60.6	41.9	22.7	43.2	50.4
+ SE (Hu et al., 2018)		39.6	62.0	43.1	23.7	44.0	51.4
+ ECA (Wang et al., 2020b)		40.3	62.9	44.0	24.5	44.7	51.3
+ RLA (Zhao et al., 2021)		41.2	61.8	44.9	23.7	45.7	53.8
+ MRLA (Fang et al., 2023)		41.3	62.9	45.0	**24.7**	45.5	53.8
+ S6LA (Ours)		**41.7**	**63.0**	**45.2**	24.6	**45.6**	**53.9**
ResNet-50 (He et al., 2016a)	RetinaNet	35.6	55.5	38.2	20.0	39.6	46.8
+ SE (Hu et al., 2018)		37.1	57.2	39.9	21.2	40.7	49.3
+ ECA (Wang et al., 2020b)		37.3	57.7	39.6	21.9	41.3	48.9
+ RLA (Zhao et al., 2021)		37.9	57.0	40.8	22.0	41.7	49.2
+ MRLA (Fang et al., 2023)		39.1	58.6	**42.0**	23.6	**43.3**	50.8
+ S6LA (Ours)		**39.3**	**59.0**	41.9	**23.7**	42.9	**51.0**
ResNet-101 (He et al., 2016a)		37.7	57.5	40.4	21.1	42.2	49.5
+ SE (Hu et al., 2018)		38.7	59.1	41.6	22.1	43.1	50.9
+ ECA (Wang et al., 2020b)		39.1	59.9	41.8	22.8	43.4	50.6
+ RLA (Zhao et al., 2021)		40.3	59.8	43.5	24.2	43.8	52.7
+ MRLA (Fang et al., 2023)		41.0	60.0	43.5	24.3	44.1	52.8
+ S6LA (Ours)		**41.2**	**60.4**	**43.8**	**24.9**	**45.1**	**53.0**

Figure: Object detection results of different methods on MS COCO2017. The **bold** fonts denote the best performance.

Results

Table: Object detection and instance segmentation results of different methods on MS COCO2017 with Mask R-CNN as a framework. The **bold** fonts denote the best performance.

Method	Params	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{m}	AP^{m}_{50}	AP^{m}_{75}
ResNet-50	44.2 M	37.2	58.9	40.3	34.1	55.5	36.2
+ SE	46.7 M	38.7	60.9	42.1	35.4	57.4	37.8
+ ECA	44.2 M	39.0	61.3	42.1	35.6	58.1	37.7
+ 1 NL	46.5 M	38.0	59.8	41.0	34.7	56.7	36.6
+ GC	46.9 M	39.4	61.6	42.4	35.7	58.4	37.6
+ RLA	44.4 M	39.5	60.1	43.4	35.6	56.9	38.0
+ MRLA	44.4 M	40.4	**61.8**	44.0	**36.9**	57.8	**38.3**
+ S6LA (Ours)	44.9 M	**40.6**	61.5	**44.2**	36.7	**58.3**	**38.3**
ResNet-101	63.2 M	39.4	60.9	43.3	35.9	57.7	38.4
+ SE	67.9 M	40.7	62.5	44.3	36.8	59.3	39.2
+ ECA	63.2 M	41.3	63.1	44.8	37.4	59.9	39.8
+ 1 NL	65.5 M	40.8	63.1	44.5	37.1	59.9	39.2
+ GC	68.1 M	41.1	63.6	45.0	37.4	60.1	39.6
+ RLA	63.6 M	41.8	62.3	46.2	37.3	59.2	40.1
+ MRLA	63.6 M	42.5	**63.3**	46.1	38.1	60.3	40.6
+ S6LA (Ours)	64.0 M	**42.7**	**63.3**	**46.2**	**38.3**	**60.5**	**41.0**

Different variants of S6LA

- The influence of $\mathbf{X}$ on h (where the opposite is h randomized for each iteration);
- The hidden state channels set to 16, 32, and 64;
- The selective mechanism involving the interval Δ and coefficient B;
- For the Transformer-based method, using simple concatenation instead of multiplication.

Results

Table: The influence of trainable h and selective mechanism of Δ and B.

Model		Params	Top-1
ResNet	S6LA	25.8 M	**78.0**
	w/o trainable h	25.8 M	77.4
DeiT-Ti	S6LA	6.1 M	**73.3**
	w/o trainable h	6.1 M	72.5
ResNet	S6LA	25.8 M	**78.0**
	w/o selective	25.8 M	77.3
DeiT-Ti	S6LA	6.1 M	**73.3**
	w/o selective	6.1 M	72.7

Table: The influence of latent dimension N and the treatment of DeiT-Ti.

Model		Params	Top-1
ResNet	$N = 16$	25.8 M	77.9
	$N = 32$	25.8 M	**78.0**
	$N = 64$	25.9 M	77.7
DeiT-Ti	$N = 16$	5.9 M	72.7
	$N = 32$	6.1 M	**73.3**
	$N = 64$	6.3 M	72.9
DeiT-Ti (S6LA)		6.1 M	**73.3**
DeiT-Ti (Concatenation)		6.1 M	72.6

Conclusion

- We have demonstrated an enhanced representation of information derived from the original data by treating outputs from various layers as sequential data inputs to a state space model (SSM).
- We propose Selective State Space Layer Aggregation (S6LA) module uniquely combines layer outputs with a continuous perspective.
- Empirical results indicate that the S6LA module significantly benefits classification and detection tasks, showcasing the utility of statistical theory in addressing long sequence modeling challenges.

Thank you!