Evolving Attention with Residual Convolutions

Yujing Wang [1 *] **Yaming Yang** [2 *] **Jiangang Bai** [1 2] **Mingliang Zhang** [1 2]
Jing Bai [2] **Jing Yu** [3] **Ce Zhang** [4] **Gao Huang** [5] **Yunhai Tong** [1]

Abstract

Transformer is a ubiquitous model for natural language processing and has attracted wide attentions in computer vision. The attention maps are indispensable for a transformer model to encode the dependencies among input tokens. However, they are learned without explicit interactions in each layer and sometimes fail to capture reasonable patterns. In this paper, we propose a novel and generic mechanism based on evolving attention to improve the performance of transformers. On one hand, the attention maps in different layers share common knowledge, thus the ones in preceding layers can instruct the learning of attention in succeeding layers through residual connections. On the other hand, low-level and high-level attentions vary in the levels of abstraction, so we adopt additional convolutional layers to capture the evolutionary process of attention maps. The proposed evolving attention mechanism achieves significant performance improvement over various state-of-the-art models for multiple tasks, including image classification, natural language understanding and machine translation.

1. Introduction

Transformer (Vaswani et al., 2017) is the state-of-the-art architecture for sequential modeling which achieves superior performances in various applications, such as natural language understanding (Devlin et al., 2019), image generation (Parmar et al., 2018) and time-series forecasting (Li et al., 2019). The performance of a transformer model mainly depends on its capability of inducing reasonable attentions between input tokens. However, as illustrated by some previous works (Tang et al., 2018; Jain & Wallace, 2019), the attention maps captured by vanilla attention layers are not always effective and explainable. To cope with this problem, recent efforts concatenated self-attentions with convolutional layers to obtain better image or text representations (Bello et al., 2019; Wu et al., 2020), whereas the attention map itself was not ameliorated. In this paper, we consider another question, *can we improve the learning of attention maps via a dedicated model architecture design?* As we will illustrate, it is possible to improve the quality of attention maps by an evolving mechanism based on a chain of convolutional modules.

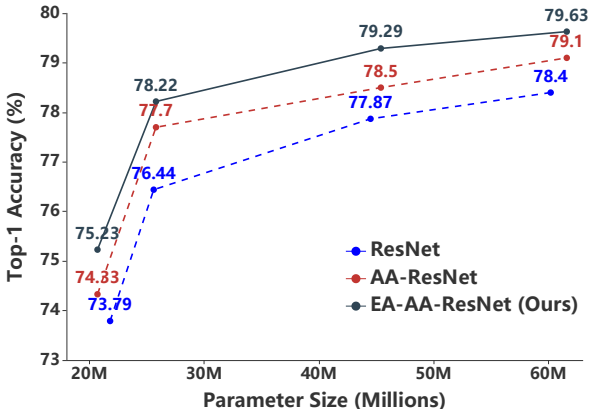


Figure 1. Comparison of different model architectures for ImageNet classification

In a vanilla transformer, the attention maps in each layer are learned without explicit interactions, which does not bring good generalization ability. Intuitively, one can simply share attention maps among layers, but it is not effective as different layers may require attention structures from different abstraction levels. For instance, in image classification tasks, a lower layer usually focuses on the relations between similar colors and textures, while a higher layer needs to reason about dependencies among different components. Our motivation is to design a dedicated module to improve the quality of attention maps in an evolutionary process. Therefore, we directly bridge the attention maps from different layers through residual connections. Moreover, we adopt convolutional layers to capture the evolution of atten-

[*]Equal contribution [1]Peking University [2]Microsoft Research [3]Institute of Information Engineering, Chinese Academy of Sciences [4]ETH Zurich [5]Tsinghua University. Correspondence to: Yujing Wang <yujwang@pku.edu.cn>, Yaming Yang <yayaming@microsoft.com>, Yunhai Tong <yhtong@pku.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, PMLR 108, 2021.

tion patterns, as this inductive bias emphasizes local details and produces more precise attention maps by reasoning on previous ones.

To this end, we propose Evolving Attention (EA-) Transformer, which guides the learning of attention maps via a chain of residual convolutional modules coupled with the transformer architecture. In each block, EA-Transformer takes all attention maps generated by the previous block as a multi-channel image. Then, with 2D-convolution over that image, the attention maps for the current block can efficiently evolve from previous ones. As such, the generic patterns of inter-token dependencies are shared across all blocks, and the attention maps are adapted to an appropriate abstraction level for each layer.

As illustrated by a case of ImageNet classification in Figure 2, the attention maps learned by EA-Transformer correctly highlight the structure of a bird with the help of convolution-based attention evolution. Especially, the residual links between attention maps facilitate the information flow of inter-token relations. Meanwhile, the convolutional module imitates an evolutionary mechanism and guides the self-attention layer to induce better attention maps. In contrast, the vanilla transformer learns each layer separately and sometimes produces vague attention structures.

We apply the generic idea of evolving attention to multiple state-of-the-art models, including Transformer (Vaswani et al., 2017), BERT (Devlin et al., 2019) and Attention Augmented (AA-) ResNet (Bello et al., 2019). Experimental results demonstrate the superiority of evolving attention for various tasks in computer vision and natural language processing domains. As shown in Figure 1, for ImageNet classification, it consistently improves the accuracy of AA-ResNet with different model capacities. AA-ResNet is a strong SOTA which encapsulates self-attentions and convolutions jointly for image representation. We also examine the generality of evolving attention in BERT-style pre-trained models and achieve consistent improvement on various kinds of backbones.

The contributions of this paper are highlighted as follows.

- We propose a novel evolving attention mechanism augmented by a chain of residual convolutional modules. To the best of our knowledge, this is the first work that considers attention maps as multi-channel images for pattern extraction and evolution, which sheds new lights on the attention mechanism.
- Extensive experiments have demonstrated consistent improvement in various natural language and computer vision tasks. As indicated by extensive analysis, both residual connections and convolutional inductive bias are beneficial to produce better attention maps.
- The proposed evolving attention mechanism is generally applicable for attention-based architectures and has further impacts on a broader range of applications.

2. Related Work

Transformer is first introduced by Vaswani et al. (2017) for machine translation and then widely adopted in numerous tasks in natural language (Devlin et al., 2019), computer vision (Parmar et al., 2018; 2019) and time-series (Li et al., 2019) domains. Transformer is solely composed of self-attention and feed-forward layers. It is much more parallelizable than Recurrent Neural Networks (RNNs) and demonstrates superiority in large-scale training scenarios. Notably, the text representation model, BERT (Devlin et al., 2019), is based on an architecture of deep bidirectional Transformer. After pre-trained on a large-scale language corpus, BERT can be fine-tuned with just one additional output layer to create state-of-the-art performance for a wide range of text-related applications.

The assumption behind Transformer is that the intra-sequence relationships can be captured automatically through self-attention. However, in practice, it is questionable if a self-attention layer learns reasonable dependencies among input tokens. Many endeavors are trying to analyze the attention maps generated by the attention mechanism. Raganato et al. (2018) analyze the Transformer model for machine translation and show that some attention heads are able to capture certain relations implicitly: lower layers tend to learn more about syntax while higher layers tend to encode more about semantics. Tang et al. (2018) suggest that the ability of inducing syntactic relations for a Transformer model is weaker than its recurrent neural network counterpart. Tay et al. (2020) argue that explicit token-token interaction is not important and dot-product attention can be replaced by synthesized attention maps. Moreover, there is a debate on whether or not the intermediate representations offered by attention mechanisms are useful to explain the reasons for a model's prediction (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019). In short, the attention maps induced by existing attention mechanisms are not good enough. Besides, there are successful attempts to combine convolutional and self-attention layers to enrich image and text representations (Bello et al., 2019; Wu et al., 2020). However, to the best of our knowledge, our work is one of the first that takes attention maps as multi-channel images and utilizes a dedicated deep neural network for pattern extraction and evolution. We believe this is a promising direction that deserves more investigations in the future.

Another limitation of Transformer lies in its prohibition for modeling long sequences, as both the memory and computation complexities are quadratic to the sequence length. To address this problem, Reformer (Kitaev et al., 2020)

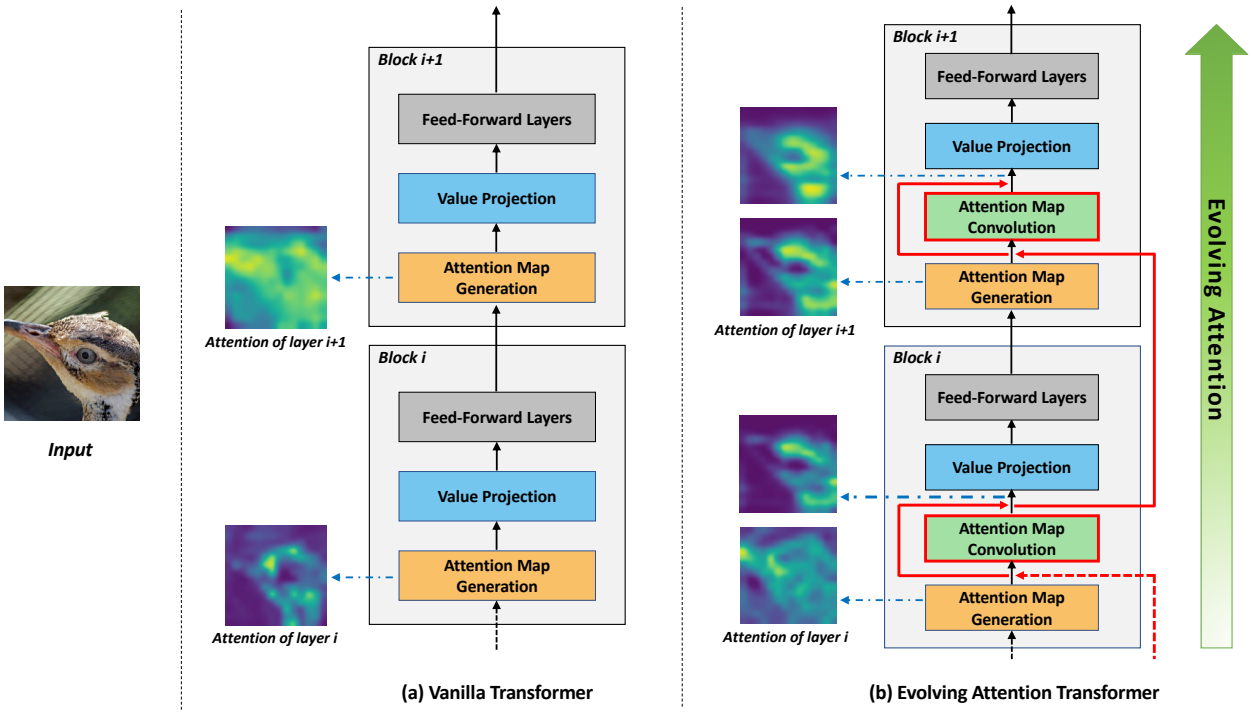


Figure 2. Architecture overview. Note that (a) is a standard transformer, where we omit layer norms and skip connections in the figure for brevity. The red lines denote residual connections, and the attention maps from the 17th and 18th blocks of an exemplar case are presented. The vanilla Transformer obtains very broad and vague attention maps at the 18th block. Instead, Evolving Attention Transformer generates reasonable attention maps at both blocks, and there is a clear evolutionary trend across successive layers.

utilizes two techniques to improve the efficiency of Transformers: (1) revising dot-product attention with locality-sensitive hashing; and (2) replacing residual layers with reversible ones. Moreover, Gehring et al. (2017) leverage an architecture based entirely on convolutional neural networks for sequence to sequence learning, where the number of non-linearities is fixed and independent of the input length. Parmar et al. (2019) apply stand-alone self-attention layers to image classification by restricting the attention operations within a local region of pixels. Vision Transformer (ViT) (Dosovitskiy et al., 2020) divides an image into a sequence of patches and utilizes an architecture as closely as possible to the text-based Transformer. Lambdanetworks (Bello, 2021) adopt lambda layers which model long-range interactions between a query and a structured set of context elements, which replaces the vanilla attention mechanism with a reduced memory and computation cost. In addition, there are other research directions, including relative positional representations (Shaw et al., 2018), adaptive masks for long-range information (Sukhbaatar et al., 2019), tree-based transformer (Shiv & Quirk, 2019), and AutoML-based evolved transformer (So et al., 2019). These works are orthogonal to ours and most of them can directly benefit from the proposed evolving attention mechanism.

3. Evolving Attention Transformer

3.1. Overview

The representation of a text sequence can be written as $\mathbf{X} \in R^{N \times C}$, where N denotes the sequence length and C is the dimension size. For an image representation, the conventional shape is (H, W, C), where H, W and C denote height, width and channel size of the image respectively. In order to apply a standard Transformer to the image representation, we flatten its shape as $\mathbf{X} \in R^{N \times C}$, where $N = HW$, and each pixel serves as an individual token in the Transformer model.

A standard Transformer block is composed of a self-attention layer and two position-wise feed-forward layers. The attention map is generated by each self-attention layer separately without explicit interactions among each other. However, as we have argued in the introduction, an independent self-attention layer does not have a good generalization ability to capture the underlying dependencies among tokens. Therefore, we adopt a residual convolutional module that generalizes attention maps in the current layer based on the inherited knowledge from previous layers. The proposed mechanism is named as Evolving Attention (EA).

A transformer architecture augmented by evolving attention

is illustrated in Figure 2(b). Each Evolving Attention (EA-) Transformer block consists of four modules, including *Attention Map Generation*, *Attention Map Convolution*, *Value Projection*, and *Feed-Forward Layers*. The residual connections between attention maps (highlighted by the red lines) are by design to facilitate the attention information flow with some regularization effects. Note that we omit layer norms in the figure for brevity. In the rest of this section, the details of each module will be introduced separately.

3.2. Attention Map Generation

Given the input representation $\mathbf{X}$, the attention maps can be calculated as follows. First, we compute the query and key matrices for each attention head through linear projections, *i.e.*, $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \mathbf{K} = \mathbf{X}\mathbf{W}^K$, where $\mathbf{Q}$ and $\mathbf{K}$ denote query and key matrices respectively, $\mathbf{W}^Q$ and $\mathbf{W}^K$ are linear projection parameters. Then, the attention map is derived by a scaled dot-product operation:

$$\mathbf{A} = \text{Attention}(\mathbf{X}) = softmax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}). \quad (1)$$

Here $\mathbf{A}$ denotes the attention map and d is the hidden dimension size. To inject sequential information into the model, we incorporate positional encoding to the input representation. The positional encoding can be either absolute or relative, and we follow the original implementation for each baseline model. The absolute positional embedding (Vaswani et al., 2017) is added to token embedding $\mathbf{X}$ directly. For relative positional representation (Shaw et al., 2018), the attention formulation can be re-written as:

$$\mathbf{A} = \text{Attention}(\mathbf{X}) = softmax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{R}), \quad (2)$$

where $\mathbf{R} = \{\mathbf{r}_{ij}\}$ is the matrix of relative positional encoding. For text data, we have $\mathbf{r}_{ij} = \mathbf{q}_i^T \mathbf{e}_{i-j}$, where $\mathbf{e}_{i-j}$ is a trainable embedding vector in terms of the relative indices for two tokens. For image data, we adopt two separate embedding vectors for height and width (Bello et al., 2019).

$$\mathbf{r}_{ij} = \mathbf{q}_i^\top \mathbf{e}^H_{h(j)-h(i)} + \mathbf{q}_i^\top \mathbf{e}^W_{w(j)-w(i)}, \quad (3)$$

where $\mathbf{q}_i$ is the query representation for the i^{th} pixel, $\mathbf{e}^H$ and $\mathbf{e}^W$ represent for trainable embedding vectors of height and width respectively, $h(i)$ and $h(j)$ are the height indices for the i^{th} and j^{th} pixels, $w(\cdot)$ denotes the index of width.

3.3. Attention Map Convolution

In a vanilla Transformer, the attention maps in each layer are calculated independently without explicit interactions between each other. Instead, in EA-Transformer, we build explicit skip connections between adjacent attention maps. Assume there are K heads in each layer. Then, we have K output attention maps from the *Attention Map Generation* module. They construct a tensor $A \in R^{N\times N\times K}$ (N is the sequence length), which can be viewed as a $N \times N$ image with K channels. Taking this as input, we adopt one 2D-convolutional layer with 3×3 kernels to generalize the attention maps. The output channel is also set to be K, so the attention maps of all heads can be generated jointly. We apply a ReLU activation after each 2D-convolution layer to provide non-linearity and sparsity. Finally, the result attention map is combined with input and fed into a softmax activation layer. Mathematically,

$$\begin{aligned}
\mathbf{A}^i_{input} &= \alpha \cdot \mathbf{A}^{i-1}_{logit} + (1-\alpha) \cdot \text{Attention}(\mathbf{X})^i_{logit}, \\
\mathbf{A}^i_{logit} &= \beta \cdot \text{CNN}(\mathbf{A}^i_{input}) + (1-\beta) \cdot \mathbf{A}^i_{input}, \\
\mathbf{A}^i &= softmax(\mathbf{A}^i_{logit}),
\end{aligned} \quad (4)$$

where $\mathbf{A}^{i-1}_{logit}$ is the attention logit matrix from the previous block; $\text{Attention}(\mathbf{X})^i_{logit}$ is the logit matrix calculated by the current self-attention block, following equation (2) without softmax; $\mathbf{A}^i_{input}$ is the combined matrix after residual connection, which serves as input to the convolutional module. CNN denotes a 2D-convolutional layer with ReLU activation. $\alpha, \beta \in [0, 1]$ are hyper-parameters for linear combination. In our experiments, the values of α and β are chosen empirically on the validation set for each task.

3.4. Value Projection and Feed-Forward Layers

Given the attention map $\mathbf{A}$, the rest of a EA-Transformer block includes value projection and position-wise feed-forward layers that are identical to a standard transformer block. The value projection layer can be formulated as:

$$\mathbf{H}_k = \mathbf{A}_k \mathbf{X} \mathbf{W}^V_k, \quad \mathbf{H} = (\mathbf{H}_1 \oplus \mathbf{H}_2 \oplus ... \oplus \mathbf{H}_K)\mathbf{W}^O, \quad (5)$$

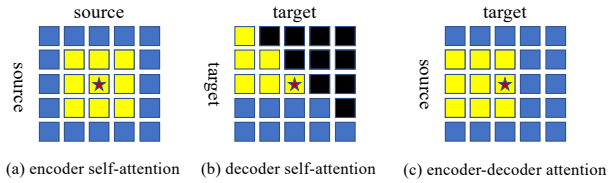
where $\mathbf{A}_k$ is the attention map for the k^{th} head, $\mathbf{W}^V_k$ is the parameter of value projection, and $\mathbf{H}_k$ is the corresponding representation generated by value projection. Afterwards, the representations of all heads are concatenated (denoted by $\oplus$) and fed into a linear projection layer with parameter $\mathbf{W}^O$. At last, the block is finished by two position-wise feed-forward layers:

$$\text{EA-Transformer}(\mathbf{X}) = \text{ReLU}(\mathbf{H}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2. \quad (6)$$

Conventionally, the dimension of $\mathbf{W}_1$ is four times of both $\mathbf{W}^O$ and $\mathbf{W}_2$, forming a bottleneck structure. We omit layer norms and skip connections here for brevity.

3.5. Convolution for Decoders

In a sequence to sequence transformer network, there are three kinds of attentions, i.e., encoder self-attention, decoder self-attention, and encoder-decoder attention. For the encoder network, we adopt a standard convolution, where



(a) encoder self-attention (b) decoder self-attention (c) encoder-decoder attention

Figure 3. Different convolution strategies for encoder and decoder

Model	#Params	#FLOPs	Top-1	Top-5
ResNet-34	21.8M	7.4G	73.79	91.43
AA-ResNet-34	20.7M	7.1G	74.33	91.92
EA-AA-ResNet-34	20.7M	7.9G	**75.23**	**92.35**
ResNet-50	25.6M	8.2G	76.44	93.19
AA-ResNet-50	25.8M	8.3G	77.70	93.80
EA-AA-ResNet-50	25.8M	8.7G	**78.22**	**94.21**
ResNet-101	44.5M	15.6G	77.87	93.89
AA-ResNet-101	45.4M	16.1G	78.50	94.02
EA-AA-ResNet-101	45.4M	17.2G	**79.29**	**94.81**
ResNet-152	60.2M	23.0G	78.40	94.20
AA-ResNet-152	61.6M	23.8G	79.10	94.60
EA-AA-ResNet-152	61.6M	25.7G	**79.63**	**94.85**

Table 1. Accuracy comparison for ImageNet classification

the surrounding pixels in a sliding window are taken into consideration. For the decoder part, we need a different convolution strategy to prevent foreseeing subsequent positions. In Figure 3, we visualize the convolution strategies for three kinds of attention maps, where the current token is identified by a star. The yellow pixels are considered by convolution, while other pixels are not included. As illustrated by Figure 3(b), the decoder self-attention only takes upper-left pixels in the convolution. The upper-right pixels (black color) are permanently masked, while other pixels (blue color) are not calculated for the current token. This results in a convolution kernel with a receptive field of 6, which can be implemented as follows: (1) performing standard 3×3 convolution with masks in the upper-right corner; (2) after convolution, shifting the entire attention matrix by 1 pixel to the bottom and 1 pixel to the right. As illustrated by Figure 3(c), the encoder-decoder attention only takes the left pixels in the convolution kernel to prevent information leakage. This can be implemented by a standard 3×3 convolution with 1 pixel shifting to the right.

4. Experiments

4.1. Image Classification

AA-ResNet (Bello et al., 2019) demonstrated that traditional CNN models could benefit from attention mechanisms in computer vision. Here we take AA-ResNet as the backbone model for ImageNet classification. AA-ResNet concatenates the image representations computed by self-attentions and convolutional neural networks. Our EA-AA-ResNet architecture enhances the attention mechanism by bridging the attention maps from different layers and extracting generic attention patterns through convolutional modules.

Model	ImageNet Top-1	Top-5
AA-ResNet-34	74.33	91.92
EA-AA-ResNet-34	**75.23**	92.35
w/o Convolution	74.34	91.98
w/o Skip Connection	74.29	91.85
with 1×1 *Convolution*	74.99	92.20
with 5×5 *Convolution*	75.12	**92.55**

Table 2. Ablation study for ImageNet classification

Settings. We follow the experimental protocol of AA-ResNet which adds self-attention to standard ResNet architectures (He et al., 2016). We leverage 10% training data to choose the hyper-parameters and perform evaluation on the development set. Specifically, we set $\alpha = 0.5$ and $\beta = 1.0$ for EA-AA-ResNet-34, while a hyper-parameter analysis and the settings for other architectures can be found in the appendix. All models are trained by 1.28 million training images for 100 epochs on 8 TESLA V100 GPUs. Finally, we report the top-1 and top-5 accuracy on 50k validation images. Major hyper-parameters are as follows: optimizer is SGD with momentum 0.9, batch size is 32 per worker, weight decay is 1e-4. For the first 5 epochs, the learning rate is scaled linearly from 0 to 0.128, and then it is divided by 10 at epoch 30, 60, 80 and 90.

Results. As shown in Table 1, AA-ResNets consistently outperform corresponding ResNet baselines by a large margin. The proposed EA-AA-ResNets further boost the top-1 accuracy by 1.21%, 0.67%, 0.80% and 0.67% on top of AA-ResNet-34, -50, -101 and -152 respectively. These numbers are statistically significant under 95% confidence level. As demonstrated in Figure 1, the performance enhancement is consistent in different model capacities. Arguably, this is owing to better attention maps induced by the proposed evolving attention mechanism. We will show more evidences in the analysis section.

Ablation Study. To understand the importance of each component, we conduct ablation experiments for the EA-AA-ResNet-34 architecture. In Table 2, *w/o Convolution* means removing the attention convolution module from EA-AA-ResNet ($\beta = 0$); *w/o Skip Connection* means removing skip connections between adjacent attention maps ($\alpha = 0$); Besides, we also replace standard 3×3 convolutions with 1×1 or 5×5 kernels. As indicated by the results, both convolutions and skip connections are crucial for the final performance. One can also notice a significant accuracy

drop using 1×1 convolutions (analogies to FFN), which indicates that a convolutional inductive bias is beneficial for capturing evolving attention patterns. Meanwhile, 5×5 convolutions also work fairly well, but we prefer 3×3 convolutions as they require fewer parameters.

4.2. Natural Language Understanding

Pre-trained language models like BERT become popular in recent years. These models are based on bi-directional transformer architectures and pre-trained by a large corpus. Our solution can be easily plugged into an existing checkpoint of vanilla BERT and achieve significant improvement through continuous training. To demonstrate this advantage, we choose GLUE benchmark (Wang et al., 2018) for an empirical study.

Settings. The encoder network of BERT consists of multiple transformer blocks. In EA-BERT, we replace each transformer block with EA-Transformer illustrated in Figure 2(b). We load the pre-trained checkpoints of BERT-Base, T5-Base, BERT-Large and RoBERTa-Large directly, and fine-tune them for each downstream task individually on the task-specific training data. The additional parameters introduced by EA-BERT are initialized randomly and trained jointly with other parameters during fine-tuning. We use the Adam optimizer (Kingma & Ba, 2014) with epsilon 1e-8. The dropout ratio is set as 0.1 empirically. Hyper-parameters are tuned in the following search space on the validation set: learning rate {1e-4, 1e-5, 2e-5}, batch size {8, 16}, training epochs {2, 3, 5}, $\alpha = \{0.1, 0.2, 0.4\}$ and $\beta = \{0.1, 0.2, 0.4\}$. The values of hyper-parameters chosen for each task will be reported in the appendix.

Results. The comparison between BERT-style models are shown in Table 3. The T5-Base and BERT-Large models are evaluated on the development set in order to be comparable with existing baselines. Other models are evaluated on the test set. EA-BERT generally performs better than vanilla BERT in different downstream tasks. Specifically, EA-BERT-Base, EA-T5-Base, EA-BERT-Large and EA-RoBERTa-Large achieve average scores of 84.4, 84.6, 85.7 and 89.4 on the GLUE benchmark, increasing 1.3, 1.1, 1.4 and 0.9 absolute points respectively compared to corresponding baselines on the development set. On the test set, the average scores are lifted by 1.0 points consistently on top of BERT-Base and RoBERTa-Large models. All improvements are achieved by loading existing checkpoints and fine-tuning the parameters with limited training time, which is an appealing advantage in large-scale applications. In addition, it is worth mentioning that EA-BERT models boost the performance largely for CoLA, indicating its superior generalization ability for small datasets. Moreover, the EA-based architectures demonstrate more benefits over two concurrent works, i.e., Realformer (He et al., 2020) and Synthesizer (Tay et al., 2020). Realformer utilizes residual connections over attention maps, but does not exploit convolution-based modules to capture evolving attention patterns. Synthesizer generates a separate attention map and mixes it with vanilla ones.

Ablation Study We perform an ablation study for EA-BERT-Base on four text datasets with different data scales. According to the results in Table 4, the privilege of EA-BERT comes from both convolution-based pattern extraction and residual connections simultaneously. Removing the convolutional module or replacing it with 1×1 kernel causes a significant performance drop. Skip connections are also advantageous, without which we only have small gains over the vanilla model. Similar to image classification, 5×5 kernel also generates competitive results.

4.3. Machine Translation

Machine translation is a common benchmark for testing sequence to sequence architectures. In our experiments, we take three machine translation datasets: IWSLT'14 German-English (De-En), WMT'14 English to German (En-De) and WMT'14 English to French (En-Fr).

Settings. In an EA-Transformer, the convolution modules are applied to encoder self-attention, decoder self-attention and encoder-decoder attention separately. Skip connections are only used in the encoder network as we find they harm the performance of a decoder. We set $\alpha = 0.1, \beta = 0.1$ for EA-Transformer-Lite and $\alpha = 0.5, \beta = 0.1$ for EA-Transformer-Base. We adopt Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and an inverse square root learning rate scheduling with linear warm-up. Warm-up step is set to be 4000 and label smoothing is 0.1. For each task, we tune the learning rate from {1e-4, 5e-4, 1e-3} and dropout ratio from {0.1, 0.2, 0.3}, which will be reported in the appendix.

We follow the code by HAT (Hardware-Aware Transformers, ACL2020) to report the number of parameters and FLOPs. The number of parameters is counted without word embedding and final softmax, so it is similar for different datasets. The number of FLOPs is estimated on a pseudo sample where the lengths of source and target tokens are both 30.

Results. We compare Transformer and EA-Transformer with different model capacities. Transformer-Lite (Wu et al., 2020) is a light architecture where all dimensions are set as 160 to replace a bottleneck structure. Transformer-Base follows the configuration in Vaswani et al. (2017), which has 6 layers for the encoder and 6 layers for the decoder network. It has 8 heads, 512 dimension for normal layers and 2048 dimension for the first FFN layer, forming a bottleneck structure. As illustratged in Table 5, EA-based models achieve consistent improvement for multiple datasets and network architectures while requiring only a few extra parameters

Model	#Params	#FLOPs	Avg	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m/-mm	QNLI	RTE
BERT-Base (dev)	109.5M	6.3G	83.1	56.3	92.7	89.9/85.6	89.5/89.7	91.3/88.3	84.5/84.5	91.3	69.3
EA-BERT-Base (dev)	110.4M	6.8G	**84.4**	**61.6**	**93.6**	**90.9/87.0**	**90.0/90.2**	**91.5/88.5**	**84.9/85.0**	**92.0**	**70.8**
BERT-Base (test)	109.5M	6.3G	79.3	52.6	93.5	87.5/82.7	84.9/83.4	**71.3/89.4**	84.6/83.8	**91.0**	68.9
EA-BERT-Base (test)	110.4M	6.8G	**80.3**	**56.7**	**93.8**	**88.1/84.1**	**86.3/85.3**	**71.3**/89.3	**85.0/84.3**	**91.0**	**69.9**
T5-Base (dev)	220.2M	9.1G	83.5	53.1	92.2	92.0/88.7	89.1/88.9	88.2/91.2	84.7/85.0	91.7	76.9
Synthesizer (dev)	272.0M	11.3G	84.1	53.3	92.2	91.2/87.7	89.3/88.9	88.6/91.4	85.0/84.6	**92.3**	81.2
EA-T5-Base (dev)	221.2M	9.9G	**84.6**	**53.7**	**93.1**	**92.3/89.0**	**89.6/89.1**	**88.8/91.9**	**85.1/85.0**	**92.3**	**81.5**
BERT-Large (dev)	335.0M	12.2G	84.3	60.5	94.9	89.3/85.4	87.6/86.5	92.1/89.3	86.8/85.9	92.7	70.1
RealFormer (dev)	335.0M	12.2G	84.8	59.8	94.0	**90.9**/87.0	**90.1/89.9**	91.3/88.3	86.3/86.3	91.9	**73.7**
EA-BERT-Large (dev)	336.7M	12.9G	**85.7**	**62.9**	**95.2**	**90.9/89.4**	89.7/88.2	**92.4/90.1**	**87.9/86.8**	**93.9**	72.4
RoBERTa-Large (dev)	355.0M	12.7G	88.5	65.8	95.9	91.2/87.8	92.1/92.0	92.2/89.6	90.2/90.1	94.9	80.7
EA-RoBERTa-Large (dev)	356.7M	13.3G	**89.4**	**68.1**	**96.8**	**92.0/88.7**	**92.3/92.2**	**92.3/89.7**	**90.6/90.4**	**95.0**	**85.3**
RoBERTa-Large (test)	355.0M	12.7G	86.3	61.2	96.2	91.4/88.5	91.2/90.5	73.3/89.6	89.8/89.4	94.8	80.0
EA-RoBERTa-Large (test)	356.7M	13.3G	**87.3**	**66.1**	**96.4**	**92.1/89.2**	**91.5/90.7**	**73.4/89.8**	**90.5/89.8**	**94.9**	**83.7**

Table 3. Comparison of different models on GLUE benchmark. The metrics of SST-2, MNLI, QNLI and RTE are accuracy. The metrics of MRPC and QQP are F1/accuracy. Matthew's correlation coefficient and Pearson/Spearman correlation coefficient are used for CoLA and STS-B respectively. The average GLUE score is calculated by the first metric of each dataset.

Model	CoLA	SST-2	MRPC	MNLI
BERT-Base	58.4	92.9	89.9/85.6	84.7/84.5
EA-BERT-Base	**61.6**	**93.6**	**90.9/87.0**	**84.9/85.0**
w/o Convolution	52.1	93.4	89.6/84.8	84.5/84.4
w/o Skip Connection	52.9	93.5	90.2/85.8	84.2/83.8
with 1×1 *Convolution*	57.5	93.5	89.8/85.5	84.7/84.8
with 5×5 *Convolution*	59.2	93.5	90.6/86.5	84.8/85.0

Table 4. Ablation study for text understanding

and computations.

Ablation Study. The ablation results of EA-Transformer-Lite are listed in Table 6. First, we remove the convolutional modules for the attention of encoder, decoder, and encoder-decoder respectively. According to the results, the convolutional module is extremely important for the encoder network. It also has positive effects on decoder self-attention and encoder-decoder attentions. Consistent with previous conclusions, replacing 3×3 convolutions with 1×1 kernels leads to a huge decay in BLUE score. Therefore, the locality inductive bias is indispensable for a transformer model to evolve towards better attention structures.

5. Analysis

5.1. Quality of Attention Maps

To examine the quality of image attention maps, we select the middle attention layers (16th, 17th and 18th) in the 34-layers network for analysis. We take the attention maps centered on the middle pixel, the shape of which is $N \times N \times K$, where $N = 14$ is the image length (after pooling) and $N = 8$ is the number of heads. We take these attention maps as direct inputs to another CNN model for classification. If the key structures are better captured in the attention maps, we should get higher accuracy scores for this task. As shown in Table 7, AA-ResNet-34 does not generate precise attention maps for the 16th and 18th layers, as the classification accuracy is relatively low. Instead, EA-AA-ResNet-34 induces good attention maps for all three layers, indicating the superiority of evolving attention.

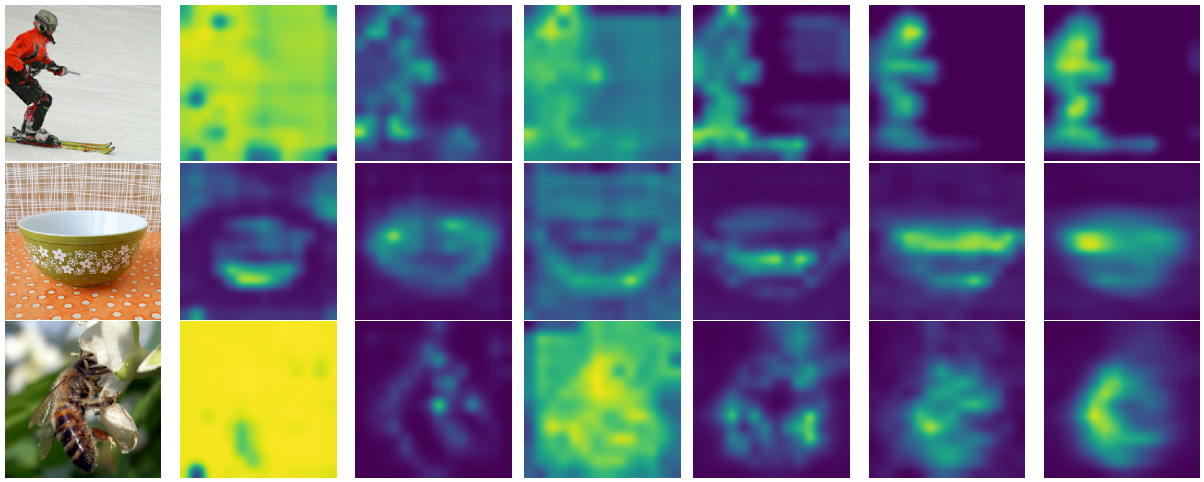
Figure 4 shows three exemplar cases for ImageNet classification, where the attention maps from representative heads of the 16th, 17th and 18th layers are visualized. These layers are at the middle of the network, which correspond to an appropriate abstraction level for visualization. Notably, AA-ResNet prefers to extract broad and vague attention patterns. In contrast, EA-AA-ResNet generates much sharper attention maps, and there exists a clear evolutionary trend in three consecutive layers. For the skier case, the attention map has successfully captured the main object in the 16th layer. Then, the outline becomes much clearer at the 17th layer with the assistance of evolving attention. Finally, the 18th layer is further improved as it identifies a complete skateboard. Other cases in Figure 4 demonstrate a similar phenomenon.

5.2. Interpretability

A good model should generate not only correct predictions, but also give faithful reasons to explain the decisions. The ERASER benchmark (DeYoung et al., 2020) is proposed to evaluate the faithfulness of rationales for text representation models. Following the original setting, we adopt BERT+LSTM and EA-BERT+LSTM as the text representation models respectively, and utilize a state-of-the-art method, LIME (Ribeiro et al., 2016) to generate rationales for each model. The experimental results are listed in Table

Model	#Params	#FLOPs (En-De)	IWSLT'14 De-En	WMT'14 En-De	WMT'14 En-Fr
Transformer-Lite	2.48M	158.31M	33.32	21.11	33.22
EA-Transformer-Lite	2.49M	163.46M	**33.80**	**21.63**	**34.12**
Transformer-Base	44.14M	2.68G	34.55	27.47	40.79
EA-Transformer-Base	44.15M	2.70G	**35.30**	**27.56**	**41.54**

Table 5. BLUE scores on machine translation datasets



(a) Input (b) AA-16 (c) AA-17 (d) AA-18 (e) EA-AA-16 (f) EA-AA-17 (g) EA-AA-18

Figure 4. Visualization of exemplar attention maps from the 16th, 17th and 18th layers of AA-ResNet-34 and EA-AA-ResNet-34 models.

Model	De-En	En-De	En-Fr
Transformer-Lite	33.21	21.11	33.22
EA-Transformer-Lite	**33.80**	**21.63**	**34.12**
w/o Encoder Convolution	32.84	21.07	33.51
w/o Decoder Convolution	33.56	21.45	33.72
w/o Encoder-Decoder Convolution	33.59	21.41	33.73
w/o Skip Connection	33.70	21.43	34.06
with 1×1 *Convolution*	33.15	21.20	33.52
with 5×5 *Convolution*	33.54	21.40	34.08

Table 6. Ablation study for machine translation

Models	16th Layer	17th Layer	18th Layer
AA-ResNet-34	24.50	31.18	17.55
EA-AA-ResNet-34	**31.02**	**31.71**	**31.93**

Table 7. Accuracy of attention map classification

Dataset	Model	Perf.↑	AUPRC↑	Comp.↑	Suff.↓
Movie Reviews	BERT	0.970	0.280	0.187	0.093
	EA-BERT	**0.975**	**0.313**	**0.194**	**0.089**
FEVER	BERT	0.870	0.291	0.212	**0.014**
	EA-BERT	**0.886**	**0.307**	**0.236**	**0.014**
MultiRC	BERT	0.655	0.208	0.213	-0.079
	EA-BERT	**0.674**	**0.221**	**0.241**	**-0.089**
CoS-E	BERT	0.487	0.544	0.223	0.143
	EA-BERT	**0.491**	**0.552**	**0.231**	**0.140**
e-SNLI	BERT	0.960	0.399	0.177	0.396
	EA-BERT	**0.969**	**0.534**	**0.445**	**0.368**

Table 8. Comparison of text representation models on ERASER benchmark. "Perf." is accuracy (CoS-E) or F1 (others), AUPRC means Area Under the Precision Recall Curve; "Comp." and "Suff." denote comprehensiveness and sufficiency metrics respectively.

8. It should be noted that higher comprehensiveness scores and lower sufficiency scores are desired. According to the results, the rationales given by evolving attention models are more precise than those generated by vanilla ones. Thus, the evolving attention mechanism not only boosts the performances of text understanding models, but also improves their interpretability through inducing better attention maps.

5.3. Learning Curve Comparison

To demonstrate the efficiency of EA-Transformer, we further compare the learning curves of Transformer-Lite and EA-Transformer-Lite on IWSLT'14 De-En dataset. As shown in Figure 5, EA-Transformer-Lite always achieves lower training loss after training the same wall-clock time, although it contains relatively 3% more FLOPs in each iteration. Finally, EA-Transformer-Lite achieves better training loss and

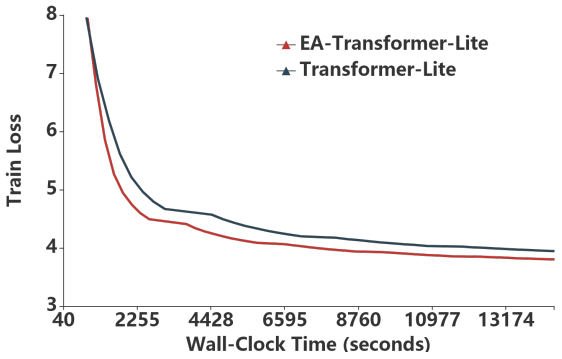


Figure 5. Learning curve comparison on IWSLT'14 De-En machine translation dataset

BLUE score at convergence.

6. Conclusion

In this paper, we propose a novel mechanism, Evolving Attention for Transformers, which facilitates the learning of attention maps via a chain of residual convolutional neural networks. It obtains superior performance in various tasks in both CV and NLP domains. Future works are considered in three aspects. First, we will apply evolving attention to more tasks and domains, such as object detection, question answering and time-series forecasting. Second, we aim to investigate other modules instead of convolutions to capture generic patterns in attention maps. Last but not least, we would like to explore bi-directional decoders that further leverage the power of convolutions.

Acknowledgement

The authors thank Haojun Jiang, Xiangtai Li and Defu Cao for helpful discussions. Gao Huang is supported in part by Beijing Academy of Artificial Intelligence.

References

Bello, I. Lambdanetworks: Modeling long-range interactions without attention. *International Conference on Learning Representations (ICLR)*, 2021.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3286–3295, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4443–4458, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In *International Conference on Machine Learning (ICML)*, pp. 1243–1252, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

He, R., Ravula, A., Kanagal, B., and Ainslie, J. Realformer: Transformer likes residual attention, 2020.

Jain, S. and Wallace, B. C. Attention is not explanation. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 3543–3556, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kitaev, N., Kaiser, L., and Levskaya, A. Reformer: The efficient transformer. In *International Conference on Learning Representations (ICLR)*, 2020.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5243–5253, 2019.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. Image transformer. In *International Conference on Machine Learning (ICML)*, 2018.

Parmar, N., Ramachandran, P., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 68–80, 2019.

Raganato, A., Tiedemann, J., et al. An analysis of encoder representations in transformer-based machine translation. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, 2016.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 464–468, 2018.

Shiv, V. and Quirk, C. Novel positional encodings to enable tree-based transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12081–12091, 2019.

So, D., Le, Q., and Liang, C. The evolved transformer. In *International Conference on Machine Learning (ICML)*, pp. 5877–5886, 2019.

Sukhbaatar, S., Grave, É., Bojanowski, P., and Joulin, A. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 331–335, 2019.

Tang, G., Müller, M., Gonzales, A. R., and Sennrich, R. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Tay, Y., Bahri, D., Metzler, D., Juan, D.-C., Zhao, Z., and Zheng, C. Synthesizer: Rethinking self-attention in transformer models. *arXiv preprint arXiv:2005.00743*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2018.

Wiegreffe, S. and Pinter, Y. Attention is not not explanation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11–20, 2019.

Wu, Z., Liu, Z., Lin, J., Lin, Y., and Han, S. Lite transformer with long-short range attention. In *International Conference on Learning Representations (ICLR)*, 2020.