Does the attention mechanism act as a correction factor?

Anonymous Author(s)

Affiliation

Address

email

Abstract

The attention mechanism has become an indispensable component in deep neural networks (DNNs). It is commonly believed that attention enhances the importance of relevant features while suppressing irrelevant ones. However, the correlation between the attention mechanism and feature importance (e.g., salience) is not always clear-cut. In essence, the optimization process in a DNN can be parsed as a recursive computation of gradients for input layers and weight parameters. ADNN precisely adheres to this paradigm, and incorporating the attention mechanism to correct input is a more straightforward approach. Additionally, we observe that the optimization process of ADNN is analogous to that of non-negative matrix factorization (NMF), where NMF conducts the iterative correction of the base matrix and coefficient matrix. In this research, we seek to correct the input layer in a manner that is analogous to correcting the base matrix following NMF optimization principles. We design a dedicated network module called the decoupled 3D self-attention module, DSM that works to simulate a correction factor for the input features. A more powerful 3D attention mechanism instead of 2D attention is utilized to mitigate the issue of high inter-class similarity inherent in medical image segmentation. Additionally, to alleviate the computational load, we begin by decoupling 3D attention into two separate components, spatial and channel attention, respectively. We then proceed to recouple these two forms of attention back together, resulting in comprehensive 3D attention. We further uncover that 3D attention can be directly transformed into the shape of 3D self-attention by introducing softmax. Besides, our DSM can easily plug-and-play into classical network backbones resulting in a NMF-like ADNN, which is a decoupled 3D self-attention network (DSNet). Our extensive experiments on Synapse, Cityscapes, ImageNet, and COCO datasets have shown that the DSNet is competent at modeling long-distance dependencies and achieves comparable performance with only a slight increase in computational costs. Our analysis points out that the attention mechanism is effective due to its capability to correct input features, which is presumably the fundamental reason for its win.

Keywords: decoupled 3D self-attention, non-negative matrix factorization, feature correction, inter-class similarity, medical image segmentation

1. Introduction

The attention mechanism [1, 2, 3, 4, 5, 6, 7, 21] has been widely employed in deep neural networks (DNNs), leading to substantial improvements in the performance of semantic

segmentation, image classification, and object detection. Despite the attention mechanism's effectiveness, there is a dearth of clear theoretical support to explain its working principle. Generally, attention is believed to be introduced to highlight significant information while suppressing unimportant information [8]. However, related studies [9, 10, 11] have shown weaker agreement between attention and feature importance. In a DNN, only the weight matrix is updated during the optimization process between two consecutive layers. Firstly, it asks whether the input layer can be updated in a manner similar to the weight matrix. Ionescu *et al*. [12] rephrase the optimization process in DNNs as a recursive calculation of gradients for input layers and weight parameters. They propose an approach for deriving the partial differential in matrix form using singular value decomposition (SVD) [13] and conducting the partial differential mapping between two consecutive layers through Matrix Backpropagation. However, the numerical stability of the matrix decomposition cannot be guaranteed when the matrix decomposition function is ill-conditioned. We have observed that an attention-based deep neural network (ADNN) [5, 6, 7] precisely adheres to this paradigm, and incorporating attention mechanisms to correct input is a more straightforward approach. Secondly, it raises the query of whether the attention mechanism plays a role in correcting the input layer. FMMNet [9] holds that the attention mechanism imposes a feature map multiplication on input features, transforming the linear piecewise function in the DNN to a high-order piecewise function. Besides, self-attention [14] or transformer [15, 16, 44, 45] models have shown promising results in capturing long-range dependencies effectively. HamNet [17] presents that the global information modeling of self-attention involves extracting the low-rank embedding of the input feature representation. This is achieved by decomposing the large matrix into low-rank matrices using techniques such as non-negative matrix factorization (NMF) [18], vector quantization (VQ) [19], or concept decomposition (CD) [20]. However, the NMF optimization algorithm is unable to save weights like a neural network which means it requires retraining during testing. Moreover, the model's sensitivity to random seeds during testing could limit its adaptability and scalability.

In this research, we investigate the connection between ADNN and NMF. We find that the optimization process of ADNN is similar to that of NMF, where NMF conducts the iterative correction of the base matrix and coefficient matrix [18]. This discovery has the potential to offer a theoretical rationale for clarifying why the attention mechanism works. The scalability and flexibility of DNNs can be attributed to their power to optimize complex functions using backpropagation, provided that network modules and annotated data are available [12]. Thus, our approach seeks to correct the input layer by designing a network module based on NMF optimization principles. This approach aligns with the working principle of the attention mechanism in ADNN and offers a more comprehensive generalization of ADNN. Finally, our approach seeks to design a specific network module to simulate the correction factor instead of directly using matrix factorization strategies such as NMF.

In the medical image task, there is often a high inter-class similarity between different categories [38] [47]. For example, the high inter-class similarity boosts the difficulty of abdominal CT organ segmentation in the Synapse dataset. Furthermore, in natural image segmentation task, such as Cityscapes, it is important to consider the impact of the inter-class similarity issue. This emphasizes the need for a more robust attention mechanism that can effectively extract discriminative features. We observe that 3D attention can effectively alleviate this issue. Besides, the extracted features for an image are denoted by $F \in \mathbb{R}^{C\times H\times W}$. In a similar fashion to how NMF

corrects the base matrix pixel-by-pixel, the attention mechanism should also be organized to perform pixel-wise feature correction, ultimately generating a 3D attention $\mathbf{A} \in \mathbb{R}^{C\times H\times W}$. In order to capture long-distance dependencies, it is necessary to obtain a 3D shape of self-attention, referred to as 3D self-attention $\mathbf{M} \in \mathbb{R}^{C\times H\times W}$, where the weights at spatial dimension sum up to 1 within the *i*-th channel $\mathbb{R}^{i\times H\times W}$. The correction operation of 3D self-attention on the input involves two main steps: feature weighting and feature aggregation. This paper then aims to decouple 3D attention into spatial and channel attention, respectively to lessen the computational cost. When the 3D attention is decoupled, the spatial and channel attention can be computed by employing BAM [22], CBAM [6], PSA [23], ECANet [41], and FcaNet [46], *etc*. Finally, the spatial and channel attention are recoupled back to form 3D attention by element-wise multiplication or addition. Self-attention mechanisms like NLNet [7] are difficult to be engaged in dense prediction tasks like semantic segmentation [23, 41, 42] since their computational complexity reaches $O(n^2)$. Fortunately, taking clues from GCNet [14], we can integrate the softmax into the whole 3D attention and directly receive 3D self-attention. Finally, this research has led us to develop the NMF-like ADNN, a decoupled 3D self-attention network, named DSNet, which is capable of figuring out 3D self-attention maps in a decoupled fashion. We have utilized DSNet for semantic segmentation, image classification, and object detection, and are pleased to report that DSNet has achieved satisfactory performance in semantic segmentation tasks, and conveyed comparable performance with the state-of-the-art models in image classification and object detection.

The contributions of this paper are as follows:

(1) In this paper, our analysis leads to the conclusion that the attention mechanism essentially corrects its input features, which may be the fundamental reason for its effectiveness. Additionally, taking clues from GCNet [14], we aim to directly convert the 3D attention into the shape of 3D self-attention by introducing softmax, which can significantly reduce the computational cost.

(2) The debate over whether convolution or self-attention is superior has always been a contentious issue. While some works [24, 2] claim that attention can entirely replace convolution operations, SegNeXt [25] points out that convolutional attention has more potential. ConvNeXt [43] gradually transformed ResNet into a transformer architecture, preserving the simplicity and efficiency of standard convolutions and this transformation ultimately resulted in surpassing the performance of the Swin Transformer [45]. This paper tries to reveal that attention or self-attention is utilized to correct input features, while convolution aims to update the weight matrix. The ultimate goal of updating the weight matrix (i.e., convolution) is equivalent to correcting the input features (i.e., attention or self-attention), despite operating on different objects.

(3) A lightweight decoupled 3D self-attention network, DSNet, is developed for semantic segmentation, image classification, and object detection tasks. DSNet demonstrates comparable performance on the Synapse, Cityscapes, ImageNet, and COCO datasets. Furthermore, the DSNet achieves faster convergence due to the simultaneous correcting input features and the updating weight matrices against the baseline.

2. Related work

Attention mechanism and Self-attention mechanism In general, attention modeling methods fall into three categories: channel attention [5], spatial attention [26, 27, 28], and hybrid attention (i.e., channel-spatial attention) [6, 22, 29, 30]. Channel-spatial attention combines both channel attention and spatial attention either in series or in parallel. Self-attention mechanism has earned

considerable interest in the realm of computer vision [31]. To reduce the high computational expense of the Non-Local Network (NLNet) [7], simplified variants of NLNet, such as EANet[32], GCNet[14], CCNet[33], Axial-DeepLab[34], are subsequently proposed. Among them, GCNet [14] finds that the global context modeled by NLNet is almost the same for various query positions within one feature map. Although channel-spatial attention combines both channel attention and spatial attention either in series or in parallel, it first applies channel attention to obtain corrected features and followed by applying spatial attention to obtain the corresponding output, which essentially belongs to 2D attention. However, in our approach, we first combine series or parallel connections to directly obtain a 3D attention form. We then correct the input features. This helps to address the problem of high inter-class similarity between different categories in medical images. To reduce the computational load, we initially decouple the 3D attention into spatial and channel attention. Finally, we recouple it back into its original 3D shape. Additionally, we can take advantage of the attribute that the self-attention map is shared among various query points. To convert the 3D attention into 3D self-attention, we can apply a softmax to the original 3D shape, thereby further reducing computational workload.

Theoretical foundation of the attention mechanism Many existing methods for explaining attention mechanisms utilize techniques like attention visualization, importance metrics, or saliency methods to identify significant features. Grad-CAM [35] is commonly used to generate visual heatmaps that explain the underlying reasoning behind the network's decision-making process. Besides, the strategy proposed by [36] utilizes deep Taylor decomposition to produce interpretability heatmaps. Those methods essentially measure the features' importance. However, several works [9, 10, 11] have revealed a weak correlation between attention maps and features' importance. This paper argues that the attention mechanism primarily corrects input features to obtain a more accurate feature representation, ultimately resulting in improved visualization performance, which may be the fundamental reason for its effectiveness.

Attention mechanism in computer vision tasks Attention mechanism has become an integral component in computer vision tasks with various variants being proposed. ECANet [41] introduces a local cross-channel interaction strategy that avoids dimensionality reduction by examining the adverse effects of the squeeze step in SENet. PSA [23] devises a polarized self-attention mechanism that effectively addresses the pixel-level regression task through polarized filtering and enhancement. FcaNet [46] utilizes frequency analysis to reparse channel attention and has demonstrated that Global Average Pooling (GAP) is a specific instance of feature decomposition in the frequency domain. Building upon this discovery, a multispectral channel attention approach is proposed. In order to extract the contextual information of the semantic segmentation model such as FCN [39], EncNet [40] introduces a new context encoding module that selectively enhances class-related feature maps and captures the semantic context. Additionally, UperNet [42] proposes a hierarchical network with a feature pyramid network to obtain an effective global prior representation. HamNet [17] utilizes strategies such as NMF [18] to decompose the large matrix into low-rank matrices and suggests that the low-rank embedding of input feature representations contains global information on self-attention. To promote the application of transformer [24] in semantic segmentation, Swin Transformer [45] introduces an inductive bias in CNN and proposes a shifted window-based design to gradually expand the receptive field. Twins-SVT [44] develops a spatially separable self-attention mechanism that can combine local and global attention. To reduce computational complexity, local attention is initially computed and grouped in the spatial dimension

and then fused to obtain global attention. However, existing ADNN models currently lack theoretical guidance in the design of attention networks, which can be considered as a black box design. In contrast, this research aims to fill this gap by incorporating NMF into the design of attention mechanisms. Finally, we present the DSNet, a NMF-like ADNN designed for semantic segmentation, image classification, and object detection tasks. Our proposed DSNet achieves comparable performance to existing advanced methods, with only a slight increase in computational costs.

3. Clarifying attention mechanism with NMF

3.1 Restating ADNN and NMF

To facilitate a clearer understanding of the relationship between NMF and ADNN, we will restate the algorithms involved in ADNN and NMF.

Attention-based Deep neural network (ADNN): Given a feature $\mathbf{F}_l$ of *the l-th* layer in the DNN, the weight matrix of the *l-th* layer is $\mathbf{W}_l$, for simplicity, ignoring the nonlinear activation here and the feature $\mathbf{F}_{l+1}$ derived from forward propagation can be written as follows, $\mathbf{F'}_{l+1} = \mathbf{F}_l \times \mathbf{W}_l$, $\mathbf{F'}_{l+1}$ needs to gradually approach $\mathbf{F}_{l+1}$, where × denotes the matrix multiplication. The weight matrix $\mathbf{W}_l$ is then updated using the stochastic gradient descent based on error backpropagation. The loss function of a DNN with L layers can be simply written as $J(\mathbf{W}) = \widehat{\mathbf{Y}} \log(\mathrm{softmax}(\mathbf{F}^{(L)}))$, where $\widehat{\mathbf{Y}}$ denotes the ground truth. The gradient concerning $\mathbf{W}_l$ can be represented as $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}_l} = \mathbf{F}_l^{\,T}\, \boldsymbol{\delta}_{l+1}$ where $\boldsymbol{\delta}_{l+1} = -\mathbf{F}_{l+1} + \mathbf{F}'_{l+1}$, so we get the final formula as $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}_l} = \mathbf{F}_l^{\,T}\,(-\mathbf{F}_{l+1} + \mathbf{F}_l \mathbf{W}_l)$. However, it is important to note that only weight matrix $\mathbf{W}_l$ is updated, while $\mathbf{F}_l$ of *l-th* layer remains untouched. In ADNN, the attention mechanism is applied to the previous layer $\mathbf{F}_l$ to obtain $\mathbf{AM}(\mathbf{F}_l)$, which is then transformed by a weight matrix $\mathbf{W}_l$ to obtain the next layer $\mathbf{F}_{l+1}$, that is $\mathbf{F}_{l+1} = \mathbf{AM}(\mathbf{F}_l) \times \mathbf{W}_l$. Ionescu *et al.* [12] rephrase the optimization process in DNNs as a recursive calculation of gradients for input layers and weight parameters. ADNN precisely adheres to this paradigm, and incorporating attention mechanisms to correct input is a more straightforward approach. Meanwhile, we find that the optimization process of NMF is similar to that of this ADNN, where NMF conducts the iterative correction of the base matrix and coefficient matrix [18]. As shown in Fig. 1, the optimization process of ADNN between any two consecutive layers is essentially a special form of solving NMF. Specifically, reweighting the input features through the attention in ADNN is equivalent to the correction of the base matrix in NMF.

Non-negative matrix factorization (NMF): In NMF theory, the primary objective is to approximate the large non-negative matrix $\mathbf{V} \in \mathbb{R}_+^{n \times m}$ by decomposing it into two low-rank non-negative matrices, namely the base matrix $\mathbf{N} \in \mathbb{R}^{n \times r}$ and the coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{r \times m}$, where r ≪ m,n. So that **NH** can approximately replace **V**, that is $\mathbf{V} \approx \mathbf{NH}$. The objective function of NMF is $J(\mathbf{V}, \mathbf{NH}) = \| \mathbf{V} - \mathbf{NH} \|_F^2$. According to the loss $J(\mathbf{V}, \mathbf{NH})$, the base matrix $\mathbf{N}$ and coefficient matrix $\mathbf{H}$ are corrected iteratively.

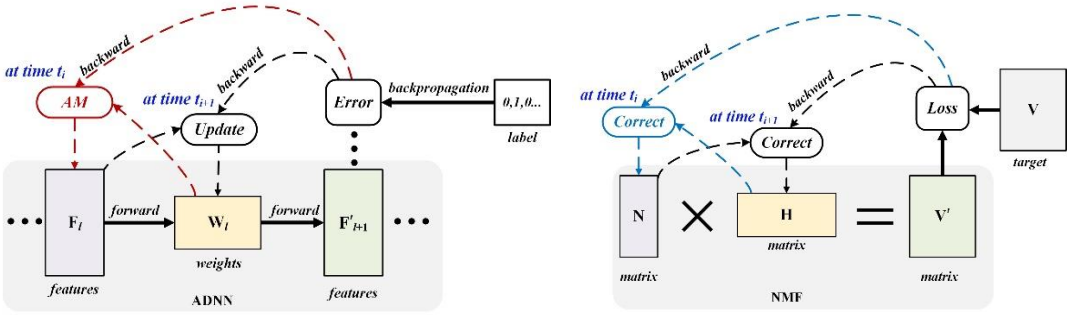

Figure 1: **Connections** between ADNN and NMF. **Left:** In the process of updating weights **W** in DNN, we take two consecutive layers as an example and observe that $\mathbf{W}_l$ is updated, but $\mathbf{F}_l$ stays unchanged during error backpropagation. However, in ADNN, we posit that the attention mechanism is utilized to update $\mathbf{F}_l$ at time t_i and $\mathbf{W}_l$ at time t_{i+1} alternately. Therefore, $\mathbf{F'}_{l+1}$ can converge to $\mathbf{F}_{l+1}$ more quickly. **Right:** In NMF, the correction process involves calculating correction parameters at time t_i to correct **N** (represented by the blue dotted line) and at time t_{i+1} to correct **H** (represented by the black dotted line).

3.2 Clarifying attention mechanism with NMF

In the context of NMF, our objective is to minimize the discrepancy between the matrix multiplication **NH** and the target **V**. To accomplish this, we utilize the stochastic gradient descent algorithm (SGD) to iteratively update the matrices **N** and **H** until the desired level of convergence is reached. Among them $\mathbf{N}' = \mathbf{N} - \alpha_1 \frac{\partial J}{\partial \mathbf{N}}$ is used to update **N**, $\mathbf{H}' = \mathbf{H} - \alpha_2 \frac{\partial J}{\partial \mathbf{H}}$ is adopted to update **H**, and $\alpha_1$and α_2 are the learning rate, respectively. To delve deeper into the process of NMF in correcting low-rank matrices, we can rephrase the updating of **H** and **N** as follows,

$$\mathbf{N}' = \mathbf{N} - \alpha_1(-\mathbf{V}\mathbf{H}^T + \mathbf{N}\mathbf{H}\mathbf{H}^T), \tag{1}$$

$$\mathbf{H}' = \mathbf{H} - \alpha_2(-\mathbf{N}^T\mathbf{V} + \mathbf{N}^T\mathbf{N}\mathbf{H}), \tag{2}$$

we can substitute $\alpha_1 = \mathbf{N}/(\mathbf{N}\mathbf{H}\mathbf{H}^T)$ and $\alpha_2 = \mathbf{H}/(\mathbf{N}^T\mathbf{N}\mathbf{H})$ into formula (1) and formula (2) resulting in formula (3) and formula (4) respectively,

$$\mathbf{N}' = \underbrace{[(\mathbf{V}\mathbf{H}^T)/(\mathbf{N}\mathbf{H}\mathbf{H}^T)]}_{\text{correction factor}} \odot \mathbf{N}, \tag{3}$$

$$\mathbf{H}' = \underbrace{[(\mathbf{N}^T\mathbf{V})/(\mathbf{N}^T\mathbf{N}\mathbf{H})]}_{\text{correction factor}} \odot \mathbf{H}, \tag{4}$$

where $\odot$ stands for element-wise multiplication. According to the presented formula (3) and (4), it is evident that NMF corrects the base matrix **N** by the correction factor $(\mathbf{V}\mathbf{H}^T)/(\mathbf{N}\mathbf{H}\mathbf{H}^T)$, and also corrects the coefficient matrix **H** by the correction factor $(\mathbf{N}^T\mathbf{V})/(\mathbf{N}^T\mathbf{N}\mathbf{H})$. The process of iterations of both matrix **N** and **H** enables fast convergence.

In DNN, SGD is employed to update the weight parameter $\mathbf{W}_l$ of l-th layer. This update can be expressed as $\mathbf{W}_l' = \mathbf{W}_l - \eta \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}_l}$, and we can rephrase the optimization process as,

$$\mathbf{W'}_l = \mathbf{W}_l - \eta\left(-\mathbf{F}_l^T\mathbf{F}_{l+1} + \mathbf{F}_l^T\mathbf{F}_l\mathbf{W}_l\right). \tag{5}$$

Meanwhile, updating weight $\mathbf{W}_l$ in formula (5) corresponds to correcting coefficient matrix **H** in NMF. Then let $\eta = \mathbf{W}_l/(\mathbf{F}_l^T\mathbf{F}_l\mathbf{W}_l)$ according to the derivation rules in NMF, the update of $\mathbf{W}_l$ as formula (5) can be converted into the corrected form of $\mathbf{W}_l$ by formula (6),

$$\mathbf{W'}_l = [(\mathbf{F}_l{}^T\mathbf{F}_{l+1})/(\mathbf{F}_l{}^T\mathbf{F}_l\mathbf{W}_l)] \odot \mathbf{W}_l, \quad (6)$$

correction factor

these two optimization rules revealed by formulas (5) and (6) are equivalent, with the only difference being in how the learning rate η is set.

So far we have concluded that correcting the weight matrix $\mathbf{W}_l$ in a DNN (as shown in formula (6)) is comparable to correcting $\mathbf{H}$ in NMF (as shown in formula (4)). However, DNN solely focuses on correcting $\mathbf{W}_l$ and doesn't address feature $\mathbf{F}_l$ of the previous layer promptly. This limitation greatly impedes the improvement of DNN performance. Similar to NMF, we suppose that the attention mechanism corrects the feature matrix $\mathbf{F}_l$ according to formula (7) which corresponds to formula (3) in NMF,

$$\mathbf{F'}_l = [(\mathbf{F}_{l+1}\mathbf{W}_l{}^T)/(\mathbf{F}_l\mathbf{W}_l\mathbf{W}_l{}^T)] \odot \mathbf{F}_l, \quad (7)$$

correction factor

here, we can deduce the formula (8) and get the gradient update of feature $\mathbf{F}_l$ from correction factor $(\mathbf{F}_{l+1}\mathbf{W}^T)/(\mathbf{F}_l\mathbf{W}_l\mathbf{W}^T)$ in a similar manner as updating weight $\mathbf{W}_l$.

$$\mathbf{F'}_l = \mathbf{F}_l - \lambda\left(-\mathbf{F}_{l+1}\mathbf{W}_l{}^T + \mathbf{F}_l\mathbf{W}_l\mathbf{W}_l{}^T\right) = \mathbf{F}_l - \lambda\left(\boldsymbol{\delta}_{l+1}\mathbf{W}_l{}^T\right), \quad (8)$$

where $\boldsymbol{\delta}_{l+1} = -\mathbf{F}_{l+1} + \mathbf{F}_l\mathbf{W}_l$.

In summary, to optimize the weight parameter $\mathbf{W}_l$, it is evident from formulas (5) and (6) that updating the weight parameter is equivalent to correcting it. Similarly, for optimizing input features $\mathbf{F}_l$, formulas (8) and (7) indicate that updating and correcting input feature $\mathbf{F}_l$ are aligned. Upon comparing formulas (6) and (7), it can be observed that the objective of correcting the weight matrix (i.e., convolution) is essentially the same as the objective of correcting the input features (i.e., attention or self-attention), albeit in different forms. This statement responds to the ongoing debate regarding the superiority of convolution and self-attention with a fresh perspective.

However, obtaining the term $\boldsymbol{\delta}_{l+1}\mathbf{W}_l{}^T$ in formula (8) directly is challenging. Due to the assumption of the local receptive field, DNN typically conducts local computations using the convolution kernel. As a result, it is not possible to obtain a large tensor $\mathbf{W}_l$. Instead, we can design a dedicated network module (i.e., our proposed DSM module in Section 4) to correct $\mathbf{F}_l$ and this module can learn a correction factor, referred to as attention, which we denote as $\mathbf{M} \approx (\mathbf{F}_{l+1}\mathbf{W}_l{}^T)/(\mathbf{F}_l\mathbf{W}_l\mathbf{W}_l{}^T)$. With $\mathbf{M}$ as the correction factor, the correction of $\mathbf{F}_l$ can be expressed as follows,

$$\mathbf{F'}_l = \mathbf{Attention} \odot \mathbf{F}_l = \mathbf{M} \odot \mathbf{F}_l. \quad (8)$$

Therefore, this research posits that the attention mechanism in ADNN essentially corrects the feature $\mathbf{F}_l$ of the input layer l. Based on the analysis of ADNN and NMF, we aim to uncover the underlying reasons for the effectiveness of the attention mechanism, including,

- The attention mechanism corrects the feature matrix $\mathbf{F}_l$, which shares the responsibility of updating weight parameters in a multi-layer neural network. This correction can potentially lead to a more optimal solution.
- By utilizing the alternating updates of $\mathbf{F}_l$ and $\mathbf{W}_l$, ADNN can accelerate model convergence (see the supplementary material) straightforwardly and effectively. It has to be mentioned that we still insist on designing specific network modules to simulate correction factors instead of the direct use of matrix factorization strategies such as NMF due to its limitations in computational costs and the inability to save weights.

4. Methodology

In this section, we design a novel module called decoupled self-attention module, DSM. The DSM module serves two important senses. Firstly, DSM approximates attention map **M** by working to simulate a correction factor for the input features **F**. Secondly, it effectively captures long-range dependencies and remarkably reduces the computational cost of self-attention.

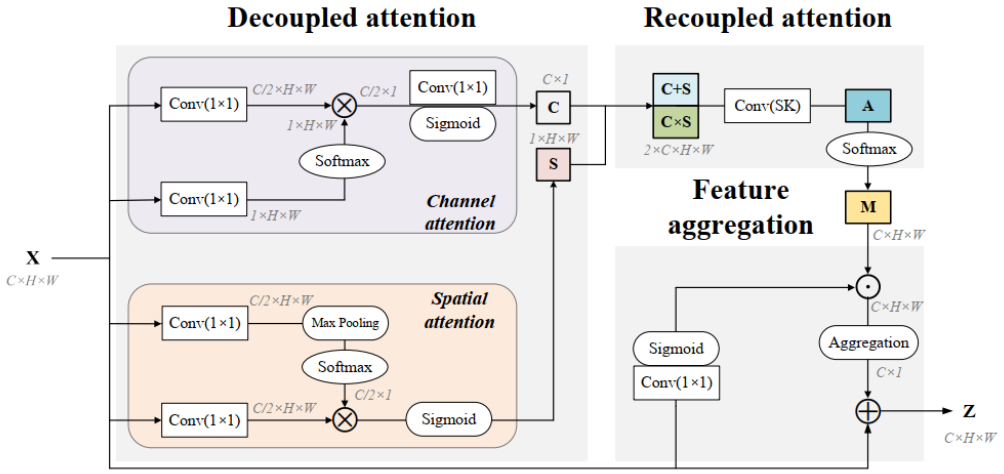

Figure 2: We propose a decoupled self-attention module DSM. Firstly, we obtain spatial attention **S** and channel attention **C** separately. Then, the 3D attention will be given back through both series and parallel recoupling operations followed by a selective kernel convolution, that is 3D attention A is obtained. Also, softmax is applied to the 3D attention A to obtain the shape of 3D self-attention M. Finally, **X** after Conv(1×1) and sigmoid will be corrected followed by feature aggregation, and then it is added element-wise to **X** in a broadcast manner to obtain Z.

4.1 Decoupling attention

Theoretically, the attention map of the 3D feature matrix should be 3D as well, but this would result in a sharp increase in computation cost. To address this issue, we begin by decoupling 3D attention into two components, spatial and channel. We then recouple these two forms of attention together, resulting in comprehensive 3D attention. The channel attention can be denoted as $\mathbf{C} \in \mathbb{R}^{C\times 1}$ and is generated with $\mathbf{C} = \sigma(Conv_{1\times1}(Conv_{1\times1}(\mathbf{X}) \times Softmax(Conv_{1\times1}(\mathbf{X}))))$, where σ represents a sigmoid function. The spatial attention can be denoted as $\mathrm{S} \in \mathbb{R}^{1\times H\times W}$ and generated with $\mathbf{S} = \sigma(Conv_{1\times1}(\mathbf{X}) \times Softmax(MaxPooling(Conv_{1\times1}(\mathbf{X}))))$.

4.2 Decoupling self-attention

The computational complexity of using NLNet to build Self-attention increases to $O((\boldsymbol{HW})^2)$, where H and W represent the height and width of the feature map, respectively. Taking inspiration from GCNet, we have obtained a 3D self-attention map by directly applying softmax to the original 3D attention map. In our proposed DSM module (as illustrated in Figure 2), we obtain spatial attention **S** and channel attention **C** separately, which are then recoupled to assemble 3D attention **A**. Then 3D attention is transformed into 3D self-attention by involving softmax on **A** to obtain **M**, that is $\mathbf{M} = Softmax(\mathbf{A})$. Subsequently, a correction factor **M** is applied to the 3D input feature $\mathbf{X}'$, where $\mathbf{X}' = \sigma(Conv_{1\times1}(\mathbf{X}))$, and feature aggregation is performed for each location to extract long-distance dependency information. The resulting output is then added element-wise to the original feature **X**. The process is formulated as $\mathbf{Z} = F_{Aggr}^{ch}(\mathrm{M} \odot \mathbf{X}') \oplus \mathbf{X} = F_{Aggr}^{ch}(Softmax(\mathbf{A}) \odot \mathbf{X}') \oplus \mathbf{X}$, where F_{Aggr}^{ch} represents the softmax operation, F_{Aggr}^{ch} defines the spatial aggregation operation for each channel, and $\oplus$ illustrates broadcast element-wise addition,

respectively.

4.3 Recoupling attention

There are two methods to recouple **C** and **S** back to yield a 3D attention map **A**. The first approach is to couple **C** and **S** in series, resulting in the multiplication of the two attentions, $\mathbf{Q} = \mathbf{C} \times \mathbf{S}$, and $\mathbf{Q} \in \mathbb{R}^{C\times H\times W}$. The second method is to recouple **C** and **S** in parallel, which is expressed as the addition of **C** and **S**, $\mathbf{P} = \mathbf{C} + \mathbf{S}$ and $\mathbf{P} \in \mathbb{R}^{C\times H\times W}$. We propose an approach that utilizes both series and parallel recoupling to their fullest potential. Building on the SKNet [37], the approach incorporates the SK module into the DSM to adaptively adjust the attention weights of **Q** and **P**, resulting in an optimized 3D attention map **A**, $\mathbf{A} = \alpha\mathbf{Q} \oplus (1-\alpha)\mathbf{P}$, where $\alpha = Conv_{SK}(\mathbf{Q}, \mathbf{P})$, and $Conv_{SK}$ denotes the selective kernel convolution, $\oplus$ denotes channel-wise addition.

5. Experiments (Further improvements will be made later)

5.1 Experimental results of semantic segmentation on the Synapse dataset

5.1.1 Experimental results compared with different attention or self-attention mechanisms on the Synapse dataset

Table 1: Experimental results compared with different attention or self-attention mechanisms on the Synapse dataset, and report p values against comparisons with baseline FCN. The p-value of our network is much less than 0.01, indicating a significant difference between our network and the baseline.

Method	**Backbone**	**mDice ↑**	**mHD95 ↓**
FCN [39]	ResNet50	81.86±0.68	26.57±5.22
FCN+SE [5]	ResNet50	82.47±0.69	24.87±3.69
FCN+EncNet [40]	ResNet50	82.08±0.26	25.00±2.41
FCN+ECANet [41]	ResNet50	81.72±0.52	23.19±1.70
FCN+CBAM [6]	ResNet50	81.82±1.01	24.38±3.46
DANet [30]	ResNet50	82.23±0.67	26.24±1.55
CCNet [33]	ResNet50	81.51±0.85	27.73±4.31
GCNet [14]	ResNet50	81.83±0.98	25.52±3.38
HamNet [17]	ResNet50	82.37±0.59	24.36±1.48
EANet [32]	ResNet50	81.77±0.25	27.00±2.50
FCN+PSA (p) [23]	ResNet50	82.66±0.64	21.99±0.88
FCN+PSA (s) [23]	ResNet50	82.48±0.56	22.33±1.79
DSNet (ours)	ResNet50	82.78±0.91	22.75±3.39
FCN+DSM (ours)	ResNet50	**83.25±0.56**	**20.55±3.57**
p-value ↓		**< 0.01**	
UPerNet [42]	ConvNeXt-B [43]	83.24±0.46	28.16±3.19
UPerNet [42]	SegNeXt-B [44]	83.86±0.38	21.98±1.83
HamNet [17]	MSCAN-B [25]	**84.72±0.51**	20.68±3.57
UperNet [42]	Swin-B [45]	84.27±0.40	22.60±1.66
UperNet [42]	Twins-SVT-B	84.09±0.24	18.92±3.02
DSNet (ours)	MSCAN-B [25]	84.69±0.50	**18.28±3.55**
p-value ↓		**< 0.01**	

Table 2 Experimental results compared with different attention or self-attention mechanisms on the Synapse dataset. (Backbones of all the methods are ResNet18).

Method	**mIoU%**
FCN[39]	66.36±0.53
FCN+SE[5]	68.40±0.35
FCN+PSA[23]	68.75±0.52

FCN+CBAM[6]	67.72±0.23
FCN+GCNet[14]	67.99±0.42
FCN+DSM(ours)	**68.97±0.05**

Table 3: Performance comparison between 2D attention VS. 3D attention.

2D attention			3D attention		mIoU(%)
Channel	Spatial	Series	Parallel	DSM	
×	×	-	-		66.36
×	✓	-	-		67.38
✓	×	-	-		68.16
✓	✓	✓	×	×	67.42
✓	✓	×	✓	×	67.94
✓	✓	✓	✓	✓	**68.97**

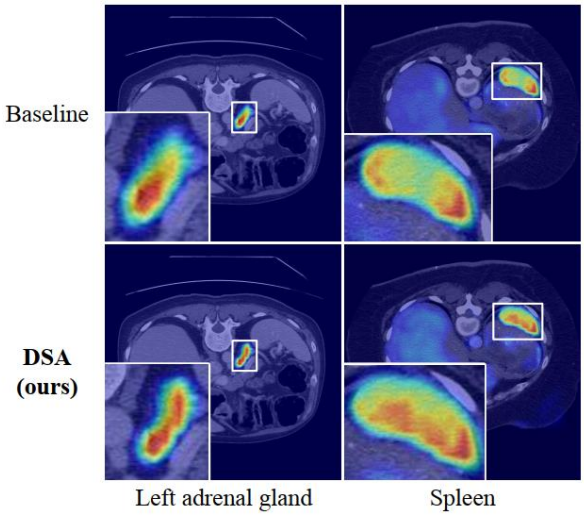

Fig. 3: Visualization of heatmaps via GradCAM. Our DSNet not only achieves better results in the "easy" category "Spleen" but also performs better in the more challenging category "Left adrenal gland" segmentation task, resulting in improved visual visualization.

5.1.2 The designed architecture in our proposed DSNet50 network

Table 4: The designed architecture in our proposed DSNet50 network. DSNet50-A is proposed to improve performance of semantic segmentation, while DSNet50-B is designed to achieve better classification and object detection results.

Method	Backbone	Plugin of every stage				Position of plugins	Ratio
FcaNet50	ResNet50	MSCA	MSCA	MSCA	MSCA*	after conv3	16
DSNet50-A	ResNet50	None	None	None	DSM**	after conv2	2
DSNet50-B	ResNet50	DSM	DSM	DSM	DSM	after conv1	16

- MSCA*: Multi-spectral channel attention
- DSM**: Decoupled self-attention module

5.1.3 Experimental results of our proposed DSNet50 compared with other networks on the

Synapse medical image dataset

Table 5: Experimental results of semantic segmentation on the Synapse dataset. Our proposed DSNet50-A achieves better segmentation results.

Method	Parameters	FLOPs	Train FPS	mDice	mIoU
ResNet50	47.13M	197.86G	7.52*	82.92	74.72
FcaNet50	49.62M	196.67G	81.60	84.06	76.1
DSNet50-A(**ours**)	50.67M	205.93G	60.4	**85.07**	**77.61**
DSNet50-B(**ours**)	48.94M	204.44G	**95.72**	83.8	75.76

*Train on a single graphics card A5000, while train on 8 graphics cards 4090.

5.2 Experimental results of semantic segmentation on the Cityscapes dataset

Table 6: Experimental results of semantic segmentation on the Cityscapes dataset. Our proposed DSNet50-A achieves better segmentation results.

Decoder	Method	Pre-trained weights*	Parameters	FLOPs	Train FPS	mIoU	mAcc	aAcc
FCN	ResNet50 V1C**	-	47.13M	395.76G	-	72.25***	-	-
FCN	FcaNet50	FcaNet50	49.65M	395.91G	48.24	75.63	82.93	95.78
FCN	DSNet50-A(**ours**)	DSNet50-A	50.68M	411.91G	46.00	**76.01**	**83.59**	**95.86**
FCN	DSNet50-B(**ours**)	DSNet50-B	48.95M	408.92G	**49.44**	75.25	83.00	95.62
FCN	ResNet50 V1C	ResNet50 V1C	47.13M	395.76G	45.6	75.51	83.19	95.89
FCN	FcaNet50	ResNet50 V1C	49.65M	395.91G	69.76	76.45	83.38	95.93
FCN	DSNet50-A	ResNet50 V1C	50.68M	411.91G	**70.96**	**77.25**	**84.48**	**96.83**

Note: Due to time constraints, we were unable to pre-train FcaNet50 and DSNet50 using ResNet V1C on ImageNet1K dataset. Instead, we employed two training strategies: (1) utilizing pre-training weights for FcaNet50 and DSNet50 based on ResNet (specifically, ResNet V1B); (2) Both models were trained using the weights of ResNet50 V1C.

**Compare to ResNet, ResNet V1C replace the 7x7 conv in the stem with three 3x3 convs.

***The mIoU results of ResNet50 are obtained using official results [3] from mmseg [2].

5.3 Experimental results of image classification on the ImageNet1K dataset

Table 7: Experimental results of image classification on the ImageNet1K dataset. Although the results of our DSNet50-B are slightly inferior to FcaNet50, our "**Train FPS**" index is significantly better, indicating that our method has a faster training speed. However, it is important to note that both our method and FcaNet50 achieve comparable results in terms of accuracy.

Method	Parameters	FLOPs	Train FPS	top1	top5
ResNet50	25.557M	4.109G	1677.51	76.48	93.17
FcaNet50	30.121M	4.112G	1036.70	**77.50**	**93.79**
DSNet50-A(**ours**)	27.369M	4.357G	**1873.95**	76.51	93.12
DSNet50-B(**ours**)	29.106M	4.208G	1424.30	77.28	93.60

5.4 Experimental results of object detection on the COCO2018 dataset

Our DSNet50-B has a smaller number of parameters compared to FcaNet. Additionally, the "**Train FPS**" index of DSNet50-B is more than twice that of FcaNet, which indicates that our network has achieved a faster training speed.

Table 8: Experimental results of object detection on the COCO2018 dataset. Although the results of our DSNet50-B are slightly inferior to FcaNet50, our "**Train FPS**" index is more than twice that of FcaNet, indicating that our method has a faster training speed.

Model	Method	Parameters	FLOPs	Train FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN	ResNet50	41.750M	187.20G	46.75	37.4	58.3	40.5	21.9	40.7	48.1
Faster-RCNN	FcaNet50	44.268M	187.31G	28.90	**38.9**	**60.2**	**42.4**	**23.1**	**42.5**	**49.9**
Faster-RCNN	DSNet50-A	45.302M	188.97G	44.95	37.8	59.4	40.8	23	41.6	48.1
Faster-RCNN	DSNet50-B	43.565M	194.27G	**64.89**	38.2	59.6	41.5	22.8	42.1	48.5

5.5 Computational time on the Synapse and ImageNet1K dataset

In the Supplementary Material, it has been observed that our DSNet50 converges with fewer iterative steps compared to the baseline ResNet50. Now, let us reconsider my method from the perspective of computational time.

5.5.1 Computational time of semantic segmentation on the Synapse dataset.

Based on the results in Fig. 4, it is evident that DSNet50-B achieves convergence in the shortest amount of time, making it a superior option compared to the ResNet50 and FcaNet50. However, DSNet50-A demonstrates a convergence speed similar to that of ResNet50. We hypothesize that a low 'Ratio' (as shown in Table 1) leads to a higher feature dimension, which in turn increases computational time in dense semantic segmentation.

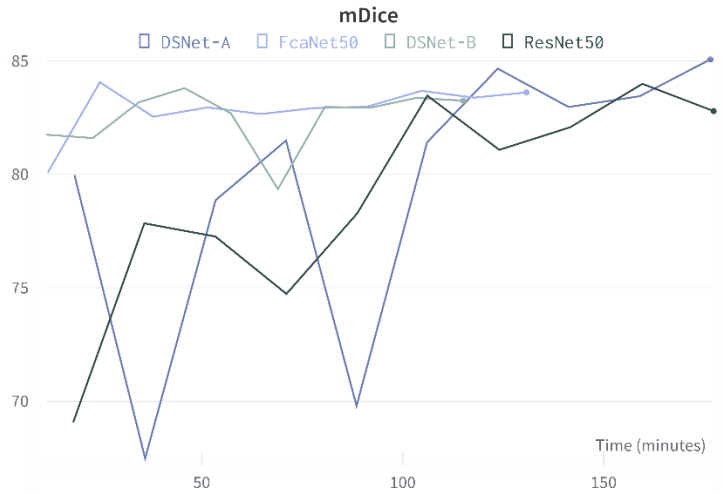

Fig. 4: Computational time of our proposed DSNet50 against compared networks on the Synapse dataset. The required computational time is ranked as follows: DSNet50-B < FcaNet50 < DSNet50-A < ResNet50.

5.5.2 Computational time of image classification on the ImageNet1K dataset.

Fig. 5 shows the required computational time is ranked as follows: DSNet50-A < ResNet50 < DSNet50-B < FcaNet50. Our proposed method, DSNet50-A, achieved optimal results in ImageNet1K dataset. Our both DSNet50-A and DSNet50-B demonstrate superior performance compared to FcaNet, indicating faster convergence performance.

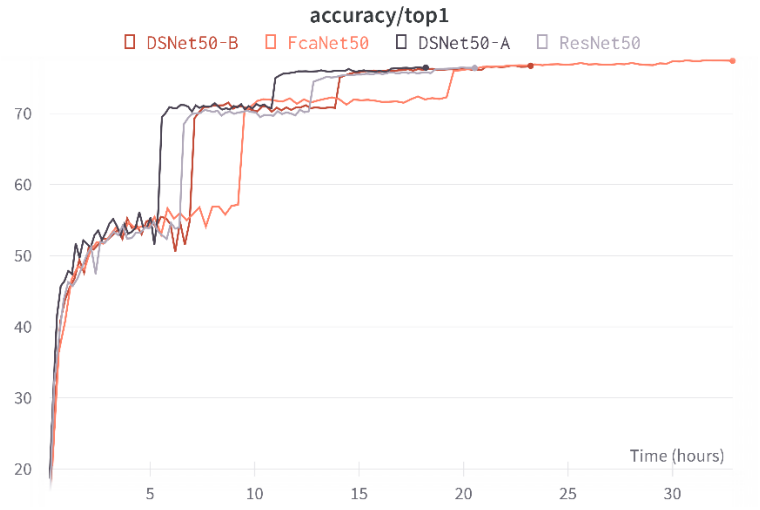

Fig. 5: Computational time of our proposed DSNet50 against compared networks on the ImageNet1K dataset. The required computational time is ranked as follows: DSNet50-A < ResNet50 < DSNet50-B < FcaNet50.

6. Limitations

Our proposed DSNet aims to simulate the correction factor (nonnegative via sigmoid) for input feature **F** through the DSM module. If we construct an additional network module to simulate the correction factor (nonnegative via sigmoid) for weight parameters **W** in formula (6), it will guarantee that the final values of **F** and **W** are non-negative as long as the initial values of **F** and **W** are also non-negative. According to the theory in the NMF, the optimization process of any two consecutive layers can be effectively explained by regarding **F** as a base matrix and **W** as a coefficient matrix. We plan to further research this strategy in the future.

In order to optimize DNN, it is crucial to adjust the learning rate dynamically. Equation (5) suggests that setting a learning rate η around $\mathbf{W}_l/(\mathbf{F}_l{}^T\mathbf{F}_l\mathbf{W}_l)$ may result in optimal performance. While this aspect has not been explored in our current research, we plan to investigate it in our follow-up work.

This study aims to explore the relationship between 3D attention and 3D self-attention. However, there is a need for further investigation into the adaptive and dynamic weight learning of cross-channel attention in the 3D self-attention mechanism. Furthermore, the specific connection between 3D attention and 3D self-attention, as well as their transformation with each other, requires further investigation.

7. Conclusion

The motivation behind the success of the attention mechanism has always been a research topic. This research suggests that by utilizing the NMF theory, the attention mechanism is able to effectively correct the input features. As a result, it leads to comparable outcomes and faster convergence against the baseline. Furthermore, our proposed DSNet is validated to have a superior visualization outcome compared to the baseline in medical and natural image segmentation. We infer that using a 3D attention mechanism instead of 2D attention can partially mitigate the issue of high inter-class similarity between different categories in semantic segmentation. Additionally, the debate on whether convolution or self-attention is superior in computer vision has been ongoing. However, our research suggests that correcting the weight matrix (i.e., convolution) is essentially

the same as correcting the input features (i.e., self-attention) albeit in different forms. It might be more suitable to integrate self-attention into the later stages of the backbone, detection, or segmentation head for dense prediction tasks.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[4] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. arXiv preprint arXiv:1803.02155, 2018.

[5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

[6] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), pages 3–19, 2018.

[7] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7794–7803, 2018.

[8] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. Computational Visual Media, 8(3):331–368, 2022.

[9] Xiang Ye, Zihang He, Wang Heng, and Yong Li. Toward understanding the effectiveness of attention mechanism. AIP Advances, 13(3), 2023.

[10] Sarthak Jain and Byron C Wallace. Attention is not explanation. arXiv preprint arXiv:1902.10186, 2019.

[11] Sofia Serrano and Noah A Smith. Is attention interpretable? arXiv preprint arXiv:1906.03731, 2019.

[12] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In Proceedings of the IEEE international conference on computer vision, pages 2965–2973, 2015.

[13] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. A practical approach to microarray data analysis, pages 91–109, 2003.

[14] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF international conference on computer vision workshops, pages 0–0, 2019.

[15] Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Lequan Yu, Liansheng Wang, and Yizhou Yu. nn-former: Interleaved transformer for volumetric segmentation. arXiv preprint arXiv:2109.03201, 2021.

[16] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. Medical transformer: Gated axial-attention for medical image segmentation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24, pages 36–46. Springer, 2021.

[17] Zhengyang Geng, Meng-Hao Guo, Hongxu Chen, Xia Li, Ke Wei, and Zhouchen Lin. Is attention better than matrix decomposition? arXiv preprint arXiv:2109.04553, 2021.

[18] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. Advances in neural information processing systems, 13, 2000.

[19] Robert M. Gray and David L. Neuhoff. Quantization. IEEE transactions on information theory, 44(6):2325–2383, 1998.

[20] Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. Machine learning, 42:143–175, 2001.

[21] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3156–3164, 2017.

[22] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. arXiv preprint arXiv:1807.06514, 2018.

[23] Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: Towards high-quality pixel-wise regression. arXiv preprint arXiv:2107.00782, 2021.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[25] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. arXiv preprint arXiv:2209.08575, 2022.

[26] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, pages 2048–2057. PMLR, 2015.

[27] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 764–773, 2017.

[28] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999, 2018.

[29] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I, pages 421–429. Springer, 2018.

[30] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3146–3154, 2019.

[31] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10076–10085, 2020.

[32] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 3531–3539, 2021.

[33] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE/CVF international conference on computer vision, pages 603–612, 2019.

[34] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV, pages 108–126. Springer, 2020.

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

[36] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. Pattern recognition, 65:211–222, 2017.

[37] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 510–519, 2019.

[38] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? IEEE transactions on medical imaging, 37(11):2514–2525, 2018.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[40] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 7151–7160, 2018.

[41] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11534–11542, 2020.

[42] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In Proceedings of the European conference on computer vision (ECCV), pages 418–434, 2018.

[43] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11976–11986, 2022.

[44] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. Advances in Neural Information Processing Systems, 34:9355–9366, 2021.

[45] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10012–10022, 2021.

[46] Qin Z, Zhang P, Wu F, et al. Fcanet: Frequency channel attention networks[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 783-792.

[47] Song Y, Teoh J Y C, Choi K S, et al. Dynamic Loss Weighting for Multiorgan Segmentation in Medical Images[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023.