sensors

MDPI

Article

Rethinking Attention Mechanisms in Vision Transformers with Graph Structures

Hyeongjin Kim and Byoung Chul Ko *

Department of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; henryjoshuakim@gmail.com
* Correspondence: niceko@kmu.ac.kr; Tel.: +82-10-3559-4564

Abstract: In this paper, we propose a new type of vision transformer (ViT) based on graph head attention (GHA). Because the multi-head attention (MHA) of a pure ViT requires multiple parameters and tends to lose the locality of an image, we replaced MHA with GHA by applying a graph to the attention head of the transformer. Consequently, the proposed GHA maintains both the locality and globality of the input patches and guarantees the diversity of the attention. The proposed GHA-ViT commonly outperforms pure ViT-based models using small-sized CIFAR-10/100, MNIST, and MNIST-F datasets and a medium-sized ImageNet-1K dataset in scratch training. A Top-1 accuracy of 81.7% was achieved for ImageNet-1K using GHA-B, which is a base model with approximately 29 M parameters. In addition, with CIFAR-10/100, the existing ViT and parameters are reduced 17-fold and the performance increased by 0.4/4.3%, respectively. The proposed GHA-ViT shows promising results in terms of the number of parameters and operations and the level of accuracy in comparison with other state-of-the-art ViT-lightweight models.

Keywords: vision transformer; graph head attention; multi-head attention; graph attention network; lightweight model

check for updates

Citation: Kim, H.; Ko, B.C. Rethinking Attention Mechanisms in Vision Transformers with Graph Structures. *Sensors* **2024**, *24*, 1111. https://doi.org/10.3390/s24041111

Academic Editor: Jing Tian

Received: 18 January 2024
Revised: 5 February 2024
Accepted: 7 February 2024
Published: 8 February 2024

1. Introduction

Transformers are among the most powerful neural network tools and have shown a promising performance using sequential data when applied to natural language processing (NLP) [1] and speech recognition [2]. A vision transformer (ViT) [3], applied in the field of computer vision, is a leading algorithm used in various vision problems, including image classification [3], image segmentation [4], object tracking [5], depth estimation [6], and action recognition [7]. ViTs are an important approach because they can be used for image processing without significantly changing the overall transformer architecture. Using a transformer with a sequence of image patches applied as an input, and without the need for a convolutional neural network (CNN), ViTs have achieved a performance exceeding that of CNN-based models. Despite such a high performance, a ViT treats all the tokens equally and ignores locality, thus losing the unique local structure of an image. In addition, ViTs still have a disadvantage in that it is difficult for them to process high-resolution images because of the large number of training data and self-attention operations. To address these problems, several studies have been conducted on improving the transformer architecture. Such studies can be divided into attempts at reducing the number of training data [8,9], operations, and parameters [10–12], as well as maintaining the locality of an image [13–15]. To reduce the number of training data, the DeiT model [8] applies a competitive convolution-free transformer with limited training data. This method relies on distillation tokens in introducing a teacher–student strategy specific to the transformer, allowing students to pay attention and learn from the teacher. Liu et al. [9] empirically analyzed various ViTs and showed that convolution with vision transformers is generally much more effective at generalization to fewer data. They also proposed a self-supervised assistant task to normalize the ViT training. A gated multi-layer perceptron (gMLP) [10] applies a simple network architecture to reduce the numbers of operations and parameters, performing as well as a transformer used in natural language and vision applications. A gMLP has experimentally achieved the same level of accuracy but without the self-attention of a ViT, thus demonstrating the lack of importance of self-attention. By replacing the self-attention sublayer with a simple linear transform that mixes with the input token, FNet [11] can accelerate the transformer encoder architecture with a limited decrease in accuracy. In particular, the training time is significantly reduced by replacing the self-attention with a Fourier transform without learnable parameters. To solve the problem of high-resolution image processing, owing to the large number of self-attention operations, PVT [12] reduces the number of operations of large feature maps using progressively size-reduced pyramids. To maintain the locality of the image, a Swin transformer [13] applies a shifted window consisting of several patches within each window, and the self-attention is calculated solely for this window. This enhances the locality of the transformer because the attention is calculated only for the region bounded by the window and not for the entire area. CeiT [15] applies an image-to-token module to extract patches from the low-level features generated instead of simple tokenization from raw input images. LocalViT [16] not only applies a depth-wise convolution to the feedforward network to reduce the computational load but also incorporates a wide range of design choices, such as an activation function, a layer placement, and an expansion ratio, as a way to provide locality mechanisms.

However, such ViT-based methods do not properly consider the spatial geometric relationship between local regions or between global and local regions and have limitations in reducing the number of computations because they depend heavily on the combination of multi-head attention (MHA). In addition, because ViTs require a large number of training data, their performance is significantly reduced when trained using small datasets without a pretraining step.

Contribution of This Work

In this study, to reduce the number of ViT operations and preserve the global and local features of the image classification, we introduce a new graph head attention (GHA) mechanism for ViT that replaces MHA with fewer graph heads using the proposed graph generation and graph attention. From this, we show that GHA works effectively on small datasets and outperforms SoTA models on conventional datasets. Our contributions are as follows:

- Unlike with other graph-based transformers [17–19], which apply graphs and attention in parallel and combine the outputs, this study is the first attempt to apply a graph inside the transformer head and replace MHA with a few GHA mechanisms. Moreover, there is no need for a class token in patch embedding, and thus the number of operations can be reduced.
- The links of nodes with low attention scores are excluded using graph pooling, and the node features are updated by applying GHA boosting to reflect the connectivity of the neighboring nodes. This process preserves the feature locality and secures the diversity of the attention. GHA-ViT not only creates tokens with local characteristics but also learns the relationship between tokens using a graph structure, which eventually strengthens the locality of the tokens.
- Because the graph structure is constructed using an attention matrix and the node feature are extracted from a value matrix, additional learning parameters for a graph construction are not required.
- GHA-ViT shows a promising classification performance with only scratch training conducted on small and medium-sized datasets and no pre-training on large datasets.

Figure 1 shows the overall architecture of the proposed GHA-ViT model.

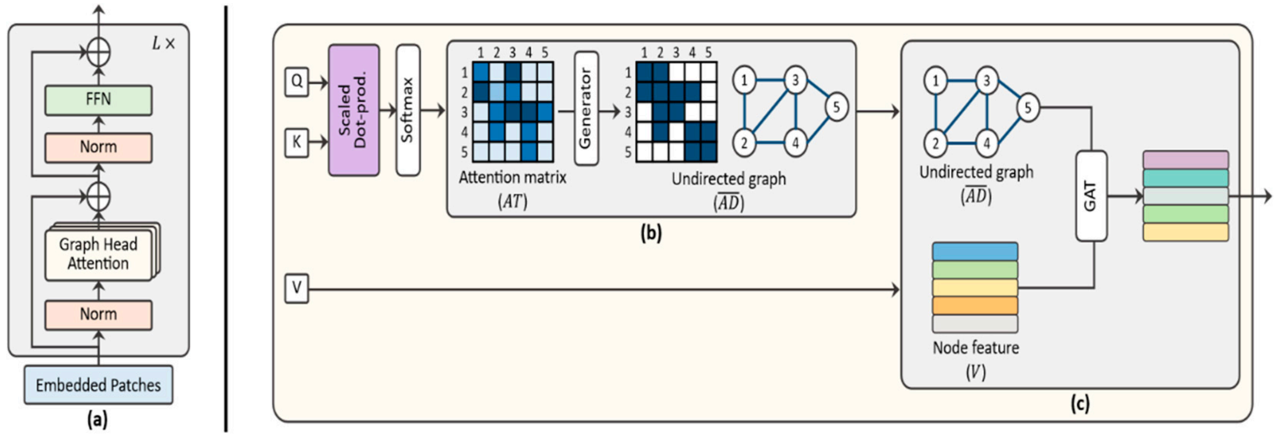

Figure 1. Overall architecture of the proposed GHA-ViT model: (**a**) GHA encoder layer composed of a few graph heads. (**b**) The attention score is calculated as the scaled dot product of Q and transposed K. Then, the graph generator is applied to the attention matrix AT for selecting sub-nodes and converted into an undirected graph $\overline{AD}$. (**c**) After graph generation, the value V and an undirected graph $\overline{AD}$ are applied to a graph attention network (GAT). Based on the generated graph and node features V, the GAT gives different importance to the neighboring nodes.

2. Related Studies

Many studies have applied graphs to transformers [20–22]. A graph transformer network (GTN) [20] was proposed to exclude noisy connections and include useful connections to tasks (e.g., metapaths) while learning effective node representations from the new graphs in an end-to-end manner to create new graph structures. Graphormer [21] utilizes a graph as a transformer to effectively encode the structural graph information into the model. With this model, centrality encoding, which can capture the node importance of the graph, and spatial encoding, which can capture the structural relationship between the nodes, are applied inside the transformer encoder. Yun et al. [22] proposed a GTN with another structure, which excludes noisy connections between graph nodes while preserving the effective nodes within the transformer. A graph-oriented transform (GraFormer) [23] was proposed for 3D pose estimation. GraFormer not only fuses the information on the graph nodes by repeatedly stacking graph attention blocks and graph convolution layer blocks but can also model the topological structure of the graphs. However, the aforementioned graph transformer models are inapplicable to vision applications because they are designed for node classification (e.g., citations, movies, traffic, social networking, and protein interactions) and human poses rather than for images.

Graph Vision Transformer Models

Unlike graph transformers being applied to general node classification, relatively few studies have applied graphs to ViTs for vision applications. Shen et al. [17] proposed a graph interactive transformer (GiT) for vehicle reidentification. Using this method, the GiT is divided into two modules: the original transformer module for extracting powerful global patch features and a local correlation graph (LCG) module for extracting local features that are distinct within the patch. The output features of the LCG and transformer modules are combined and used for downstream tasks. Zheng et al. [18] proposed a graph transformer network, which is a graph representation of an entire slide image, as well as a method for fusing transformers. The image patches become a set of nodes in the graph, and each node is connected by an edge. The constructed graph is put into the transformer using a graph convolutional network (GCN) and a pooling layer. The output of the transformer is applied to the MLP head for classification. Mesh Graphormer [19] integrated graph convolution with self-attention as a way to reconstruct human poses and meshes from a single image. It was proven that both the graph convolution and grid features of Mesh

Graphormer helped improve the performance of a pure transformer. The vision graph neural network (ViG) [24] was the first to combine graph structures with images. It regards each image patch as a single graph node and employs k-NN to build the relations between each image patch. Despite the novelty of this approach, it has the disadvantage of capturing only the similarities of the image patches without considering the latent image structure. Graph and optimization-based heterogeneous structured pruning (GOHSP) [25] applied graphs to optimize the model structure. However, because the interest was only in making the model structure more lightweight, the learning limitations of the model, such as the inductive bias, have not been improved.

In both research categories, a graph is not directly applied to the attention, and the existing attention head and graph are combined. In this case, frequent feature dimension scaling occurs for the feature combination, and the output is not a true combination of the global and local features. In addition, because MHA is used as is and a graph is added as a separate module, the number of operations can be increased. The remainder of this paper is organized as follows. Section 3 provides a brief description of the ViT and graph neural networks. We then present the details of the proposed GHA-ViT in terms of the architecture, training, and testing in Section 4. Section 5 describes the experimental environment and provides a comprehensive evaluation of the proposed method based on the results of various experiments. Finally, some concluding remarks are given in Section 6.

3. Preliminaries

3.1. Vision Transformers

Inspired by the success of the transformer scaling used in NLP, the ViT [3] directly applies a standard transformer to an image with minimal modifications. To achieve this, the ViT splits the image into P patches (such as a token in NLP) with a feature dimension of d and feeds a sequence of linear embeddings of these patches as input to the transformer. After the embedded patch matrix $H \in \mathbb{R}^{P\times d}$ is normalized, it passes through the MHA module and is fed to the MLP to generate the output of the encoder. The encoder consists of L layers and is iteratively operated from 1 to L. Finally, the representation y is obtained using calculations using the prediction head weight $W_c \in \mathbb{R}^{d\times|C|}$ and bias $b_c \in \mathbb{R}^{|C|}$, where $|C|$ is the number of classes.

3.2. Patch Generation

Pure ViTs apply a single convolutional-layer-based image tokenizer to place a given image into a small patch. They divide an image into several patches according to the given patch size without a replacement. In this regard, each patch has fundamental difficulty in securing the locality of the image. It is therefore necessary to learn a significant number of datasets to understand the inductive bias. In this study, we employed a pooling-layer-based tokenizer [26] to overcome this limitation. The pooling layer, which is helpful in enhancing the inductive bias, aids in escaping the big data paradigm.

3.3. Graph Attention Networks

A graph attention network [27] is a convolution-style neural network that can be applied to graph-structured data using a masked self-attentional layer. Unlike a GCN [28,29], a GAT assigns different importance to nodes in the same neighboring group and can therefore improve the model capacity and help with the interpretation. This structure enables inductive learning because such learning is possible without access to the entire graph. The attention score redefines the input data by determining the importance of the neighbor of node i as follows:

$$h_i' = \sigma(\alpha_{ii} h_i W + \textstyle\sum_{j\in\mathcal{N}(i)} \alpha_{ij} h_j W) \tag{1}$$

where $\mathcal{N}(i)$ is the number of nodes neighboring node i, α is the normalized attention score, and $W \in \mathbb{R}^{d\times d'}$ is the trainable parameter for the node feature $h \in \mathbb{R}^{n\times d}$ (n is the number of nodes). σ is the activation function.

4. A Graph-Head-Attention-Based ViT

4.1. Graph Head Attention

The attention head is the core idea of a transformer and calculates the attention between each patch using three matrices: the query (Q), key (K), and value (V). The attention score is calculated as the scaled dot product of Q and the transposed K, and it can be executed in parallel. It therefore has an advantage over previous neural networks in terms of its computational speed. The transformer uses MHA; thus, n heads can learn different types of attention from the input and obtain a strong attention representation by combining them [3]. In reality, however, not all the heads of the MHA have the same effect on the attention performance of the transformer. Instead, only a part of the head affects the performance, and the remainder focuses on unnecessary parts that negatively affect the outcome of the final attention [30]. From this perspective, it is clear that MHA is inessential for a transformer. In this section, we propose a new transformer that can receive more attention with fewer GHA mechanisms and without the use of multiple heads.

Let X be the input of the encoder layer. Here, X consists of P patches, and the hidden dimension of the input patch is d_h. The i-th patch can then be denoted as $x_i \in \mathbb{R}^{d_h}$ and $X = [x_1, x_2, \cdots, x_p], X \in \mathbb{R}^{P\times d_h}$. The Q, K, and V matrices have corresponding weight matrices W_Q, W_K, and $W_V \in \mathbb{R}^{d\times d_h}$, and Q, K and V can be obtained from the dot product of input X and the weight matrices. The attention matrix (AT) of the head is calculated as follows:

$$\begin{gathered} Q = XW_Q \\ K = XW_K, \quad Q, K, V \in \mathbb{R}^{P\times d_h} \\ V = XW_V \end{gathered} \tag{2}$$

$$AT = softmax\left(\frac{QK^T}{\sqrt{d_h}}\right) \in \mathbb{R}^{P\times P} \tag{3}$$

4.2. Graph Structure Generation

To provide better generalization and performance in a CNN, the pooling layer plays an important role in reducing the feature map size and broadening the receptive field. However, this pooling operation cannot be applied directly to a graph because there is no local information between the graph nodes. Therefore, inspired by the approaches described in [31,32], we propose graph pooling containing local information based on a mask filter. Graph pooling allows us to downsample the graph data and adaptively selects a subset of nodes to form new, smaller graphs. In this respect, we apply the $Top-k$ function to the attention matrix AT to select sub-nodes with significant connectivity. Sparse distilled nodes can be regarded a new form of graph structure derived from an attention matrix. In other words, we consider the sparse matrix of these nodes to be the adjacency matrix AD of a graph. In the case of a structure with a total number of patches P and a total number of layers L, the value of k selected for each layer is increased by $(0.8 \times P)/L$ from a minimum of 10% to a maximum of 80% of the total number of patches.

$$AD = Top-k(AT, k) \tag{4}$$

We can now construct a graph consisting of node patches using AD. To consider the self-edge of the node, AD adds an identity matrix I. However, as shown in Figure 2a, directed and undirected edges are mixed into the initial graph constructed from AD. In a transformer, because the attention is created according to interactions with neighboring patches, a directed edge cannot guarantee the correct patch attention. Therefore, the mixed graph must be converted into an undirected graph. For this purpose, the following graph transformation method is proposed: First, as indicated in Equation (5), the upper matrix of AD and its transpose are added to form a partially undirected graph, as shown in Figure 2b:

$$AD_{triU} = triU(AD) + triU(AD)^T \tag{5}$$

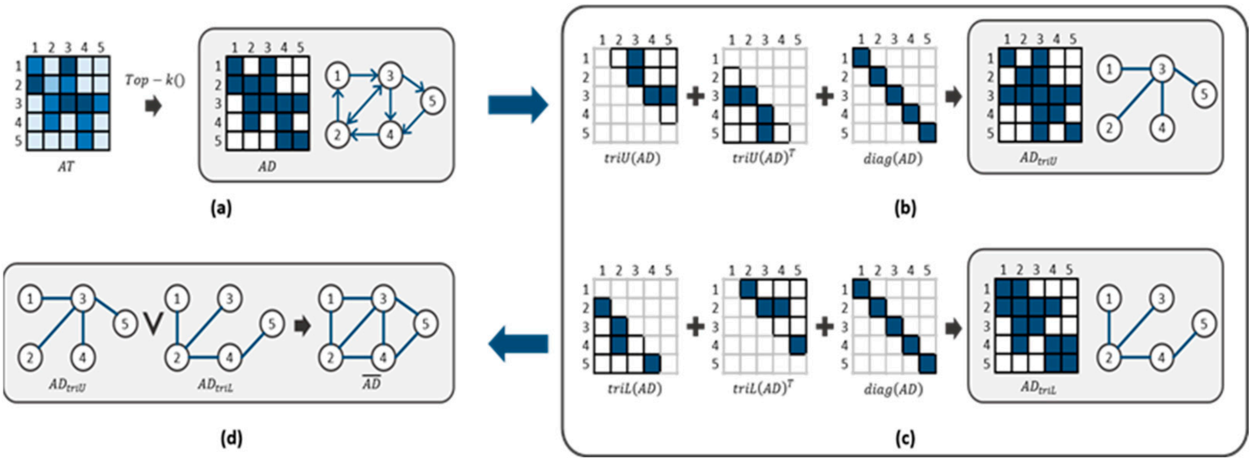

Figure 2. Graph generation process from attention matrix AT: (**a**) initial adjacency matrix and graph with threshold applied, (**b**) upper matrix AD_{triU} and its graph, (**c**) lower matrix AD_{triL} and its graph, and (**d**) undirected graph $\overline{AD}$. $\vee$ indicates OR operation.

Similarly, the lower matrix of AD and its transpose are added to form a partially undirected graph, as shown in Figure 2c:

$$AD_{triL} = triL(AD) + triL(AD)^T \tag{6}$$

Finally, the upper matrix AD_{triU} and the lower matrix AD_{triL} generate an undirected graph $\overline{AD}$ according to an OR operation $\vee$, as shown in Figure 2d.

$$\overline{AD} = AD_{triU} \vee AD_{triL} \tag{7}$$

In addition, we employ a graph calibration process to render the graph robust to irregular connectivity. The graph intensity AD_I and graph weight AD_w are applied as follows:

$$\overline{AD} := \left(\overline{AD} + \overline{AD_I}\right) \odot \overline{AD_w} \tag{8}$$

where $AD_I, AD_w \in \mathbb{R}^{P \times P}$ is a trainable parameter and $\odot$ indicates a Hadamard product.

4.3. Graph Head Attention Boosting

To improve the accuracy of the image classification, detailed attention can be obtained using a MHA combination. However, MHA requires the weight matrices W_Q, W_K, and W_V for each head. Therefore, as the number of heads increases, more learning parameters and a greater memory and computational time are required. In addition, because the attention of each head becomes similar when the number of heads exceeds a certain criterion, removing several attention heads during the test does not significantly affect the performance [30]. To avoid the problems caused by MHA, we used fewer and demonstrated that the transformer can operate successfully using only the proposed GHA. To ensure the diversity of attention, similar to MHA with fewer attention heads, and to emphasize the correlation between the graph nodes, we apply a GAT [27] to GHA. When using GHA instead of a general MHA mechanism, securing the diversity of the attention is an extremely important part of the successful operation of the GHA model. We previously extracted an $\overline{AD}$ representing the relationship between the node patches from the attention matrix. Herein, we apply the GAT to $\overline{AD}$ to achieve efficient attention computations between the nodes with $\overline{AD}$.

From $\overline{AD}$ and V, the attention coefficient e between the nodes i and j is obtained using the learnable weight matrix W_c.

$$e_{ij} = \text{FN}\left(W_c \cdot v_i, W_c \cdot v_j\right) \tag{9}$$

Here, an FN is a simple single-layer feedforward neural network that transforms the input into $\mathbb{R}^{1\times d_h} \times \mathbb{R}^{d_h\times 1} \to \mathbb{R}$. The above expression indicates the importance of the features of the nodes i-j. At this time, j does not indicate all the nodes, only the neighbors $\mathcal{N}(i)$ of node i. Finally, if it passes the softmax function, the following normalized attention matrix $\widetilde{AD}$ can be calculated:

$$\widetilde{AD}_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(e_{ik}\right)} \tag{10}$$

By applying Equation (10) to all the encoder layers, GHA can dynamically consider the connections between the nodes to ensure the diversity of the node features. The final GHA is produced by applying the node feature matrix V and the weight matrix W_{gat} of the GAT to $\widetilde{AD}$ as follows:

$$\text{GHA}(V) = \sigma_2\left(\widetilde{AD} \cdot V \cdot W_{gat}\right) \tag{11}$$

where σ_2 is a ReLU activation function. The value of GHA is obtained through the process of Figure 1c. Because the GHA encoder consists of L layers, V is put into the first layer, but from the second layer, V is changed to h_l, the output of each layer. GHA(h_{l-1}) of the previous layer is again skip-connected (element-wise sum) with input h_{l-1}.

$$h_l' = \text{GHA}(h_{l-1}) + h_{l-1}, l \in \{1\cdots L\} \tag{12}$$

After h_l' is linearly normalized (LN) again and applied to the FFN, it is similarly skip-connected to the original h_l' to produce the output of the final encoder block, as shown in Figure 1a.

$$h_l = \text{FFN}\left(h_l'\right) + h_l',\ l \in \{1\cdots L\} \tag{13}$$

Finally, the output of the last encoder layer L, $h_L \in \mathbb{R}^{P\times d_h}$, is passed to the readout layer. For the readout layer sequence pooling (seq) [26], the mean and max values were used to consider the diversity [33]. The multi-readout feature H_{out} is calculated:

$$H_{out} = h_{seq} \parallel h_{mean} \parallel h_{max} \tag{14}$$

where $\parallel$ denotes the concatenation operation between the node features. The multi-readout feature H_{out}, which has passed through the readout, is then classified using an MLP. The loss function was optimized using soft distillation [34,35].

5. The Dataset and Experimental Results

To evaluate the representation learning ability of the proposed GHA-ViT model, we compared it with VGG-16 [36], ResNet [37] and MobileNetV2 [38], which are representative CNN models; ViT-based methods [3,8,26,39]; MLP-based approaches [10,40,41]; and a graph-based method [24,25]. We proved through our experiments that the performance of the proposed model is similar to that of other state-of-the-art (SoTA) methods on several benchmark datasets. In addition, we demonstrated the effect of the graph-generating module through ablation experiments and demonstrated that a promising performance can be achieved even if fewer heads are used through attention visualization.

5.1. The Experimental Setup

Datasets. Various benchmark classification datasets are used to measure the capacity of the model. CIFAR-10/100 [42], MNIST [43], and MNIST-Fashion [44] are used as small-sized datasets, and ImageNet-1K [45] is used as a medium-sized dataset. Auto-Augment [46], Rand-Augment [47], and random erasing [48] were used as data augmentation methods.

Baseline. We set the baseline of the GHA differently to prove that the proposed GHA-ViT model can reduce the number of head and encoder layers. The basic structure of the GHA-ViT is based on DeiT [8] because it is inherently capable of learning with a small dataset. The baseline models are of two types, GHA-Base and GHA-Small, according to the number of heads and layers. Table 1 summarizes the GHA model used as a baseline. In addition to GHA-ViT, we used ResNet [37] as the CNN baseline model for the comparative experiments. The ResNet model has a modified last MLP layer to suit the number of classes for each experimental dataset. A pure ViT and DeiT are used as the transformer models for comparison with the GHA-ViT. These methods also have changed last MLP layers to obtain suitable outputs for each set of experimental data.

Table 1. Details on GHA-ViT model variants. "dim d" means hidden dimensions in encoders, and "mlp Ratio" means a scaling factor for hidden dimensions of MLP. In GHA-*-L/β, L means number of layers and β is patch size. * means model size (S: small, B: baseline).

Model	Patch Size (β)	Head	Layers (L)	dim d	mlp Ratio
GHA-S-7/3	3 × 3	3	7	64	2
GHA-B-7/3	3 × 3	6	7	64	2
GHA-S-14/7	7 × 7	3	14	64	4
GHA-B-14/7	7 × 7	6	14	64	4

5.2. The Experiment Environment and Parameter Settings

We built a GHA-ViT model using the PyTorch framework. NVIDIA 3090Ti was used for the training. The following parameters were used in the model training. AdamW was used as the optimizer, and β1 = 0.9 and β1 = 0.999 were set as the values. The batch size was set to 128 for all training and testing. The initial learning rate was set to 0.005, and cosine warm-up and decay were used. The epoch was set to 300 for training, the input resolution of ImageNet was 224 × 224, and a 32 × 32 resolution was used for the small-sized datasets.

5.3. Comparing the Performance with State-of-the-Arts Models

Table 2 presents the performance comparison results of the GHA-ViT model for the small datasets when applying scratch training SoTA methods. GHA-ViT outperformed the SoTA methods for all types of small datasets. In addition, the GHA-ViT demonstrated efficient operation in terms of the learning parameters and operations. GHA-S, which is a small version of the proposed GHA-ViT, outperformed the SoTA methods on CIFAR-10 and CIFAR-100. GHA-B showed a 0.1% lower value than ResNet-18 on the MNIST dataset and a 0.1% lower value than ViT12/4 on the MNIST-F dataset but achieved the best performance for the CIFAR-10 and CIFAR-100 datasets. Owing to a low inductive bias, transformer-based models [3,26] achieve an overall low performance in an environment with small datasets. Compared with ViG-Ti [24], the proposed GHA-B also shows a higher accuracy of 3.8% for CIFAR-10, 6.0% for CIFAR-100, 0.3% for MNIST, and 0.3% for MNIST-F. These results confirm that the proposed GHA-ViT models perform well on small-sized datasets by maintaining the high locality and globality of the images when using the proposed graph structure. In terms of the number of parameters and operations, GHA-S uses approximately 17-fold fewer parameters than pure ViT-12/4 and approximately 5.8-fold fewer operations but shows a 4.3% higher accuracy on the CIFAR-100 dataset. GOHSP [25] demonstrates a 0.6% higher performance than the proposed GHA-B. However, unlike GHA-B, which applies only scratch training, GOHSP uses a pre-trained ViT model; therefore, a difference of 0.6% indicates that GHA-B can achieve a good performance without pre-training. In addition, GOHSP requires 5 M more parameters and 1070 additional units of computational complexity than GHA-S. In comparison with SAL-ViT [49], the proposed GHA also achieved a higher performance in the trade-off between computational volume and performance. This shows that the proposed GHA uses graphs to preserve the locality information more strongly and consume less computational costs than the attention approximation method. From the overall results, we can see that the proposed GHA-ViT method guarantees an inductive bias as high as that of a conventional CNN method and that it operates efficiently, even with a small number of training data.

Table 2. Performance comparison of scratch-trained CNN and transformer-based models on small-sized datasets. Image resolution is the same at 32 × 32.

Model	Params (M) ↓	MACs (M) ↓	Top-1 (%) ↑			
			CIFAR-10	CIFAR-100	MNIST	MNIST-F
VGG-16 [36]	20	155	90.1	70.7	99.7	94.6
ResNet-18 [37]	11	40	90.2	66.4	99.8	94.7
ResNet-34 [37]	21	80	90.5	66.8	99.7	94.7
ResNet-56 [37]	24	130	93.9	71.5	99.7	94.8
ResNet-110 [37]	43	260	94.1	72.6	99.7	95.1
MobileNetV2/0.5 [38]	1	10	84.7	56.3	99.7	93.9
MobileNetV2/2.0 [38]	8	20	91.0	67.4	99.7	95.2
ViT-12/4 † [3]	85	5520	94.8	74.1	99.6	95.4
ViT-Lite-7/16 †† [26]	3	20	78.4	52.8	99.6	93.2
ViT-Lite-7/8 †† [26]	3	60	89.1	67.2	99.6	94.4
ViT-Lite-7/4 †† [26]	3	260	93.5	73.9	99.7	95.1
CVT-7/8 [26]	3	60	89.7	70.1	99.7	94.5
CVT-7/4 [26]	3	250	94.0	76.4	99.7	95.3
CVT-7/3 × 2 [26]	3	1290	95.0	77.7	99.7	95.1
ViG-Ti [24]	6	1230	93.0	74.1	99.4	95.0
GOHSP ††† [25]	10	2020	97.4	-	-	-
SAL-ViT [49]	5	1601	95.9	77.6	-	-
GHA-S-7/3	5	950	95.2	78.4	99.5	95.2
GHA-B-7/3	10	2130	96.8	80.1	99.7	95.3

†: Pure ViT consists of 12 multi-heads and 16 kernel sizes. However, because the image size is 32 × 32, the size of the kernel is reduced (to 4) in the same way as using other methods. ††: The ViT-Lite version used a different patch size on a 32 × 32 size input image. Experimental results of the ViT-Lite version are referenced from [26]. †††: The GOHSP used pre-trained weights from training on ImageNet.

We conducted experiments using the ImageNet-1K dataset to demonstrate that the proposed model works effectively on small- and medium-sized datasets. Table 3 presents the results of a performance comparison between the proposed GHA-ViT and other SoTA methods. Compared with ResNet-152, among the CNN-based methods [37], the number of parameters in the GHA-S model is reduced up to six times over, and the number of operations is reduced up to 6.4 times over. In terms of accuracy, the GHAS model is slightly inferior to ResNet-152, whereas the GHA-B model improves the Top-1 accuracy by 3.4%. In a comparison with ViT-based methods [3,8,14,26,39], in terms of accuracy, the GHA-S model has an increased Top-1 accuracy compared to that of PoolFormer-S12 under the same conditions (Param and MACs). In the case of GHA-B, the Top-1 accuracy was the highest at 81.7%; however, the number of operations was 1.1-fold higher than that of T2T-ViT-14 with a similar accuracy. Compared with MLP-based models [10,40,41], both the GHA-S and GHA-B models increased their Top-1 accuracy by approximately 2% on gMLP-Ti and gMLP-S, which have similar numbers of parameters and operations. In addition, in comparison with ViG methods [24] using graph structures, the numbers of parameters and operations are slightly higher, whereas the GHA-S and GHA-B models show high accuracies of 3.8% and 1.3%, respectively. This is because the proposed GHA-ViT model can generate a graph structure with a higher efficiency than the graph generation method used in ViG. When the GHA-B models are compared with GOHSP [25], the GHA models show a 1.8% higher Top-1 accuracy. In the case of the GHAS model, the Top-1 accuracy was 2.5% lower than that of GOHSP; however, the parameters and MACs were improved by 1 M and 1 G in comparison with GOHSP, respectively. In terms of the parameters and computational complexity, the proposed model is sufficiently competitive with other CNN and ViT-based approaches.

Table 3. Performance comparison of scratch-trained CNN and transformer-based models on the ImageNet-1K dataset. Image resolution is the same at 224 × 224.

Model	Params (M) ↓	MACs (G) ↓	Top-1 (%) ↑	Top-5 (%) ↑
ResNet-50 [37]	26	4.3	76.2	95.0
ResNet-101 [37]	45	7.9	77.4	95.4
ResNet-152 [37]	60	11.6	78.3	<u>95.9</u>
ViT-S-16 [3]	47	10.1	78.1	-
DeiT-S [8]	22	4.6	79.8	95.0
CCT-14/7 × 2 [26]	22	18.6	80.6	-
T2T-ViT-14 [14]	22	4.8	81.5	-
PoolFormer-S12 [39]	12	1.8	77.2	-
PoolFormer-S24 [39]	30	3.0	80.3	-
Mixer-B /16 [40]	59	12.7	76.4	-
ResMLP-12 [41]	15	3.0	76.6	-
gMLP-Ti [10]	6	1.4	72.3	-
gMLP-S [10]	20	4.5	79.6	-
ViG-Ti [24]	7	1.3	73.9	92.0
ViG-S [24]	23	4.5	80.4	95.2
GOHSP [25]	11	2.8	79.9	-
GHA-S-14/7	10	1.8	77.4	93.5
GHA-B-14/7	29	5.9	<u>81.7</u>	95.8

5.4. Ablation Studies

Graph Structure Generation. To verify the efficiency of the proposed graph generation method, we observed the difference in accuracy according to the change in the graph structure on various datasets, as shown in Table 4. Here, $\mathcal{G}_{w/o}$ creates a graph using only the Top − k method from the attention map, and $\mathcal{G}_{triU}$ creates a graph using an upper triangular matrix. In addition, $\mathcal{G}_{triL}$ creates a graph using a lower triangular matrix. When $\mathcal{G}_{all}$ generated graphs considering both the upper and lower triangular matrices, we obtained a higher accuracy than in the other three cases for all the datasets. As an interesting aspect of the experiment, the performance when only $\mathcal{G}_{w/o}$ was used was higher than when only $\mathcal{G}_{triU}$ and $\mathcal{G}_{triL}$ were applied. This is because a complex graph structure maintains the association between the tokens without artificially breaking them. From the experiments, we observed that if we create a graph structure by considering only specific regions of the attention map, the graph cannot accurately reflect the overall characteristics of the image.

Table 4. Comparison of accuracy (%) according to the methods of graph structure generation.

Dataset	$\mathcal{G}_{w/o}$	$\mathcal{G}_{triU}$	$\mathcal{G}_{triL}$	$\mathcal{G}_{all}$
CIFAR-10	93.2	93.1	92.8	<u>95.2</u>
CIFAR-100	73.8	73.7	73.0	<u>78.4</u>
MNIST	99.4	99.3	99.4	<u>99.5</u>
MNIST-F	94.8	94.7	94.8	<u>95.2</u>

Effect of the Multi-Readout Feature. To prove the effectiveness of the multi-readout feature, we conducted a comparison experiment with various readout methods, as shown in Table 5. Table 5 indicates that the proposed multi-readout feature exhibits the highest performance for all the small datasets. From the results, we can see that the combined multi-readout features help classify the final image more effectively than using only the readout features obtained using simple operations, such as max or average operations.

Type of Graph Convolution. Table 6 presents the results when a representative graph convolution was applied to GHA-ViT. The GCN [28], GIN [50], and GAT [27] were used as graph convolution methods. As shown in Table 6, the proposed GHA-ViT method does not exhibit a large difference in the various graph convolution operations. This indicates that the GHA inherently has a highly flexible structure. Among the three graph convolutions, the highest accuracy was obtained when a GAT was used because a GAT assigns a different importance to patches in the same neighbor group, unlike a GCN or GIN, which treat all the relationships between patches in the same manner. In this study, a GAT was used as the default graph convolution when considering the experiment results.

Table 5. Effect of the multi-readout feature. Comparison of accuracy (%) of different readout layers on small-sized datasets.

Dataset	Mean	Max	Seq	Multi
CIFAR-10	93.9	92.5	91.1	95.2
CIFAR-100	73.8	72.1	75.1	78.4
MNIST	99.2	99.4	99.2	99.5
MNIST-F	94.8	93.4	94.9	95.2

Table 6. Accuracy (%) difference for different types of graph convolution. The results from GHA-S on small-sized datasets.

Dataset	GCN	GIN	GAT
CIFAR-10	94.5	94.9	95.2
CIFAR-100	76.7	77.1	78.4
MNIST	99.4	99.4	99.5
MNIST-F	94.8	94.9	95.2

5.5. Graph Visualization

To simplify describing the suitability of the proposed graph structure generation method, we visualized the graph structure generated from the GHA-ViT based on a small (S) model. Figure 3 shows the adjacency matrix created by the GHA-S and the graph generation method of a ViG [24], which generates an adjacency matrix by directly applying k-NN to the input image patches. As shown in Figure 3, the graph structure created using k-NN includes background patches that are unrelated to the target patch. When using the proposed graph generation method, only patches that have a strong relationship with the target patch were selected. This indicates that a graph generation method using an attention map is more effective than an image-based k-NN method. In addition, when we check the tokens generated by the proposed method, we can see that the tokens around the target token are selected. This indicates that the proposed graph generation method can preserve the feature locality.

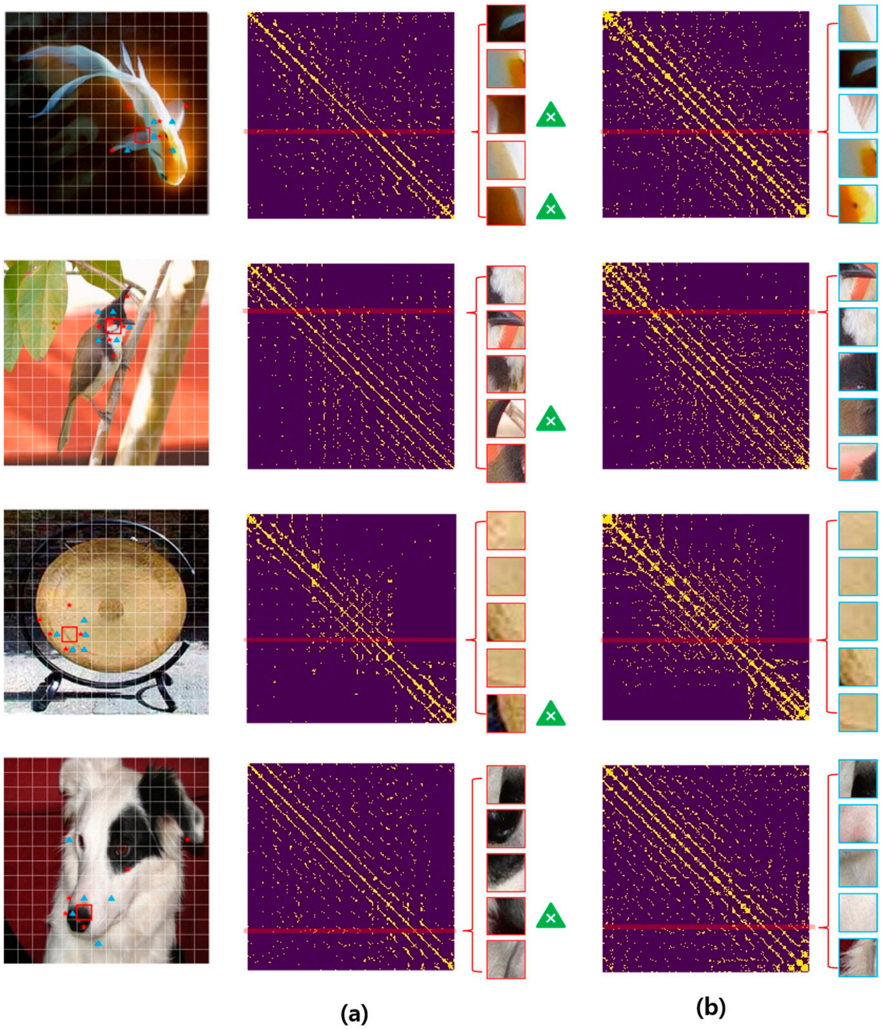

Figure 3. Graph structure visualization generated using the proposed graph generation method: (**a**) adjacency matrix generated using k-NN of the ViG [24] and (**b**) adjacency matrix using the GHA-S. The red line on the adjacency matrix indicates the row corresponding to the target patch (red star and box area) of the image. Patches on the right side of the adjacency matrix are patches that have the highest relationship with the target patch (yellow triangle and box area). As can be seen in Figure 3, the proposed method selects only patches that have a high relationship with the target patch, whereas the adjacency matrix generated using the k-NN method includes background images that have a low relationship with the target patch.

6. Conclusions

In this paper, we proposed a new GHA method that can overcome the limitations of MHA, the core module of ViTs. By converting the attention map operation from a matrix perspective into a graph perspective, it is possible to significantly reduce the number of unnecessary operations and parameters while maintaining the accuracy of the image classification. We also demonstrated that the attention feature space embedded into multiple heads differed insignificantly from that when fewer graph heads were used. Through scratch training and experiments using various small datasets, the proposed GHA-ViT demonstrates promising results without being significantly affected by the number of datasets. In future studies, to improve the mask filter when constructing a graph and achieve a more meaningful attention output, we plan to apply a method combining different graph-pooling approaches such as graph U-NET [31]. Through such additional studies, the ViT performance of the GHA structure is expected to be significantly improved in comparison with that of MHA-based ViT approaches.

Author Contributions: H.K. was responsible for the design and overall investigation. B.C.K. was responsible for the data curation, supervision, and writing and editing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Scholar Research Grant of Keimyung University in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 1–11.
2. Pham, N.-Q.; Nguyen, T.S.; Niehues, J.; Müller, M.; Stüker, S.; Waibel, A.H. Very deep self-attention networks for end-to-end speech recognition. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 66–70.
3. 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 16 × 16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021; pp. 1–21.
4. Yan, X.; Tang, H.; Sun, S.; Ma, H.; Kong, D.; Xie, X. After-unet: Axial fusion transformer unet for medical image segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2022; pp. 3971–3981.
5. Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; Wei, Y. Motr: End-to-end multiple-object tracking with transformer. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 659–675.
6. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 12179–12188.
7. Chen, J.; Ho, C.M. Mm-vit: Multi-modal video transformer for compressed video action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2022; pp. 1910–1921.
8. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 10347–10357.
9. Liu, Y.; Sangineto, E.; Bi, W.; Sebe, N.; Lepri, B.; Nadai, M. Efficient training of visual transformers with small datasets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 23818–23830.
10. Liu, H.; Dai, Z.; So, D.; Le, Q.V. Pay attention to mlps. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 9204–9215.
11. Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; Ontanon, S. FNet: Mixing tokens with Fourier transforms. In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL), Seattle, WA, USA, 10–15 July 2022; pp. 4296–4313.
12. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 568–578.
13. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 10012–10022.
14. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F.E.; Feng, J.; Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 558–567.
15. Yuan, K.; Guo, S.; Liu, Z.; Zhou, A.; Yu, F.; Wu, W. Incorporating convolution designs into visual transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 579–588.
16. Li, Y.; Zhang, K.; Cao, J.; Timofte, R.; Van Gool, L. Localvit: Bringing locality to vision transformers. *arXiv* **2021**, arXiv:2104.05707.
17. Shen, F.; Xie, Y.; Zhu, J.; Zhu, X.; Zeng, H. Git: Graph interactive transformer for vehicle re-identification. *IEEE Trans. Image Process.* **2023**, *32*, 1037–1051. [CrossRef] [PubMed]
18. Zheng, Y.; Gindra, R.H.; Green, E.J.; Burks, E.J.; Betke, M.; Beane, J.E.; Kolachalama, V.B. A Graph-Transformer for Whole Slide Image Classification. *IEEE Trans. Med. Imaging* **2022**, *41*, 3003–3015. [CrossRef]
19. Lin, K.; Wang, L.; Liu, Z. Mesh graphormer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 12939–12948.
20. Yun, S.; Jeong, M.; Kim, R.; Kang, J.; Kim, H.J. Graph transformer networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 1–11.

21. Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. Do transformers really perform badly for graph representation? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 28877–28888.
22. Yun, S.; Jeong, M.; Yoo, S.; Lee, S.; Sean, S.Y.; Kim, R.; Kang, J.; Kim, H.J. Graph transformer networks: Learning meta-path graphs to improve gnns. *Neural Netw.* **2022**, *153*, 104–119. [CrossRef] [PubMed]
23. Zhao, W.; Wang, W.; Tian, Y. Graformer: Graph-oriented transformer for 3d pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 20438–20447.
24. Han, K.; Wang, Y.; Guo, J.; Tang, Y.; Wu, E. Vision gnn: An image is worth graph of nodes. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; pp. 1–13.
25. Yin, M.; Uzkent, B.; Shen, Y.; Jin, H.; Yuan, B. Gohsp: A unified framework of graph and optimization-based heterogeneous structured pruning for vision transformer. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Washington, DC, USA, 7–14 February 2023; pp. 1–9.
26. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the big data paradigm with compact transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Virtual, 19–25 June 2021; pp. 1–18.
27. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–12.
28. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017; pp. 1–14.
29. Lee, J.; Jeong, M.; Ko, B.C. Graph convolution neural network-based data association for online multi-object tracking. *IEEE Access* **2021**, *9*, 114535–114546. [CrossRef]
30. Michel, P.; Levy, O.; Neubig, G. Are sixteen heads really better than one? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 1–11.
31. Gao, H.; Ji, S. Graph u-nets. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019; pp. 2083–2092.
32. Lee, J.; Lee, I.; Kang, J. Self-attention graph pooling. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019; pp. 3734–3743.
33. Kim, J.-Y.; Cho, S.-B. Electric energy consumption prediction by deep learning with state explainable autoencoder. *Energies* **2019**, *12*, 739. [CrossRef]
34. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. In Proceedings of the Advances in Neural Information Processing Systems Workshops (NeurIPSW), Montréal, QC, Canada, 8–13 December 2014; pp. 1–9.
35. Wei, L.; Xiao, A.; Xie, L.; Zhang, X.; Chen, X.; Tian, Q. Circumventing outliers of autoaugment with knowledge distillation. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 608–625.
36. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2019; pp. 1–14.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
38. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 4510–4520.
39. Yu, W.; Luo, M.; Zhou, P.; Si, C.; Zhou, Y.; Wang, X.; Feng, J.; Yan, S. Metaformer is actually what you need for vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 10819–10829.
40. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 24261–24272.
41. Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5314–5321. [CrossRef] [PubMed]
42. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009; pp. 1–58.
43. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
44. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
45. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A largescale hierarchical image database. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.

46. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation policies from data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 113–123.
47. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Virtual, 14–19 June 2020; pp. 702–703.
48. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020; pp. 13001–13008.
49. Zhang, Y.; Chen, D.; Kundu, S.; Li, C.; Beerel, P.A. SAL-ViT: Towards Latency Efficient Private Inference on ViT using Selective Attention Search with a Learnable Softmax Approximation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 4–6 October 2023; pp. 5116–5125.
50. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? In Proceedings of the International Conference on Learning Representations (ICLR), Neu Orleans, LA, USA, 6–9 May 2019; pp. 1–17.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.