Attention Mechanism in Computer Vision: Current Status and Prospect

Chengzhi Wu

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
chengzhi.wu@kit.edu

Abstract

As the key component in Transformer models, attention mechanism has shown its great power in learning feature relations even under long ranges in the natural language processing domain. Its success has also inspired researchers to apply it for computer vision tasks in recent years. In a variety of visual benchmarks, transformer-based models perform similar to or better than other types of neural networks such as convolutional and recurrent networks. In this report, we review the current status of the application of attention mechanism in computer vision tasks. In addition to categorizing the attention-based methods, since most current works are done with 2D image input and only a few focus on 3D data, we also propose research ideas in which attention mechanism is used for 3D data.

1 Introduction

Before Transformer was developed, recurrent neural networks (RNNs), e.g., GRU [7] and LSTM [16], were used in most state-of-the-art language models. However, RNNs require the information flow to be processed sequentially, which hinders the potential of parallel computation for faster sequence processing.

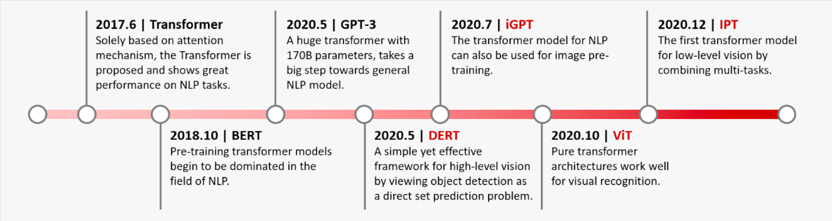

2017.6 | Transformer
Solely based on attention mechanism, the Transformer is proposed and shows great performance on NLP tasks.
2018.10 | BERT
Pre-training transformer models begin to be dominated in the field of NLP.
2020.5 | GPT-3
A huge transformer with 170B parameters, takes a big step towards general NLP model.
2020.5 | DERT
A simple yet effective framework for high-level vision by viewing object detection as a direct set prediction problem.
2020.7 | iGPT
The transformer model for NLP can also be used for image pre-training.
2020.10 | ViT
Pure transformer architectures work well for visual recognition.
2020.12 | IPT
The first transformer model for low-level vision by combining multi-tasks.

Figure 1.1: Key milestones in the development of Transformer. Models marked in red are Transformer-based models designed for CV tasks. Source from [15].

In 2017, Vaswani et al. [26] proposed Transformer, a novel encoder-decoder architecture built solely on multi-head self-attention mechanisms and feed-forward neural networks. Compared to RNNs, this attention-based model allows massive parallel computation, which enables training on larger datasets and subsequently promotes the development of large scale pre-trained models for NLP tasks. Following the pioneer work of [26], Devlin et al. [8] introduced a new language representation model named BERT to pre-train a Transformer on unlabeled text. Later, GPT-3 [1], as a massive pre-trained Transformer-based language model with 175 billion parameters, achieved astounding performance on various NLP tasks even without requiring any further fine-tuning.

On the other side, in the computer vision (CV) domain, before 2020 Convolutional neural networks (CNNs) were regarded as essential fundamentals in and almost dominate the learning methods in CV tasks. However, recent Transformer-based visual models show that they are competitive alternatives for those tasks. For image input tasks, early visual Transformer models use CNNs for first-stage latent feature computation and then apply attention methods on learned latent features, e.g. DETR [2] for image detection and DANet [12] for image segmentation. Shortly after, it is demonstrated that applying the attention mechanism solely on images is also possible for both supervised tasks, e.g. ViT [9] for classification, and self-supervised tasks, e.g. iGPT [5] for image generation. To deal with low-level vision tasks, IPT [4] combines multi-tasks in an attention-based framework and achieves great performance. Figure 1.1 shows some milestone works in the development of Transformer models, in both NLP domain and CV domain.

The key component of these Transformer models is the attention mechanism, which learns feature relations between sequence elements and even allows capturing long-term information and dependencies between them. From the feature exploitation perspective, convolution operations and attention operations are just two different ways of learning feature relations in the latent space. The former one usually focuses on local features, while the latter one usually focuses on long-range relations. There is also one interesting argument as: in our world, we discovered convolution operation before the attention operation; but probably in another parallel universe, we discovered the attention operation before the convolution operation. See more discussion in Section 3.

The rest of the report is organized as follows. Section 2 revisits the attention mechanism used in Transformers. Section 3 reviews some attention-based models for CV tasks, both on 2D data and 3D data. Since only a few works of them focus on 3D data, we propose two possible research ideas in Section 4. Finally, a conclusion is given in Section 5.

2 Revisiting the Attention Mechanism in Transformer

A Transformer consists of an encoder module and a decoder module with several encoders/decoders of the same architecture. Each encoder and decoder is composed of a self-attention layer and a feed-forward neural network. In the self-attention layer, let $X \in \mathbb{R}^{n \times d}$ denote a sequence input with n entities $(x_1, x_2, \ldots, x_n)$, where d is the embedding dimension of each entity. Multiplying by three different learnable weight matrices ($W^Q \in \mathbb{R}^{d \times d_q}$, $W^K \in \mathbb{R}^{d \times d_k}$, $W^V \in \mathbb{R}^{d \times d_v}$), each input entity x_i is first transformed into three different vectors: the query vector q_i, the key vector k_i, and the value vector v_i. They (or at least q_i and k_i) share a same dimension number, i.e., $d_q = d_k$. With a sequence of multiple entities as the input, vectors derived from different input entities are packed together into three different matrices Q, K and V. Then the output $Z \in \mathbb{R}^{n \times d_v}$ of a self-attention layer is given by:

$$Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \qquad (2.1)$$

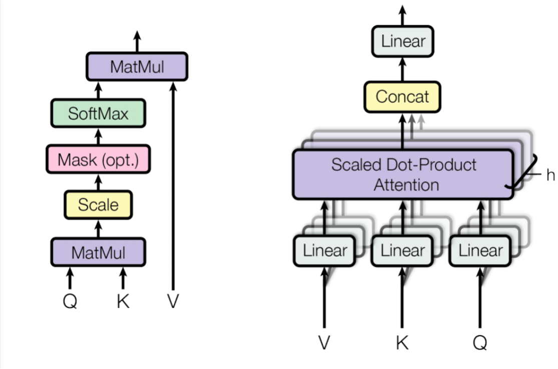


Figure 2.1: Left: self-attention. Right: multi-head attention. Source from [26].

In a more detailed explanation, Eq. 2.1 formulates the following operations. For each input entity with an embedding of vector x_i, firstly, scores between it and all other input entities are computed. Then the scores get scaled and converted into probabilities. Finally, the value vector of each input entity multiplies with the corresponding probabilities, and their sum vector z_i is the output of x_i at this layer. Additionally, similar to CNN that each convolutional layer may have multiple convolution kernels, Transformers can also use multi-head attention, as illustrated in Figure 2.1.

In recent years, researchers start to apply Transformer models on computer vision tasks, either with single-modal input or multi-modal input. Since Transformers are originally used for NLP tasks, it is quite reasonable to use them in models with both vision input and text or speech input. However, in this report, we are more interested in applying the attention mechanism on vision input, e.g. images and point clouds. Hence, models involving text or speech input will not be discussed specifically in the following sections.

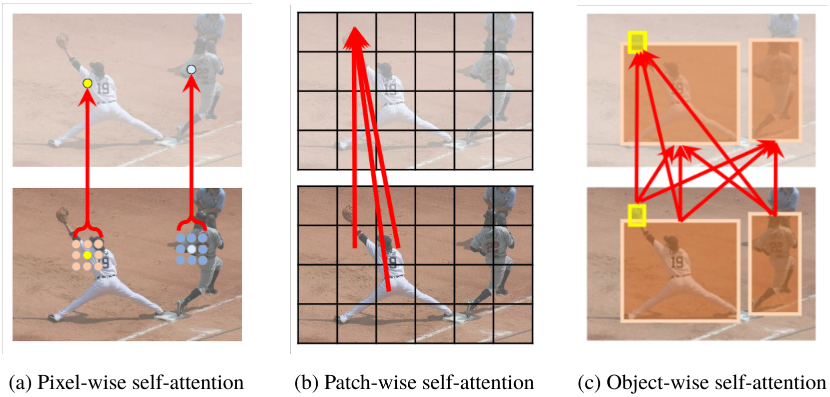

(a) Pixel-wise self-attention
(b) Patch-wise self-attention
(c) Object-wise self-attention

Figure 3.1: Different attention types based on the different input entities from images: (a) pixel-wise self-attention; (b) patch-wise self-attention; (c) object-wise self-attention.

3 Attention Mechanism in Vision

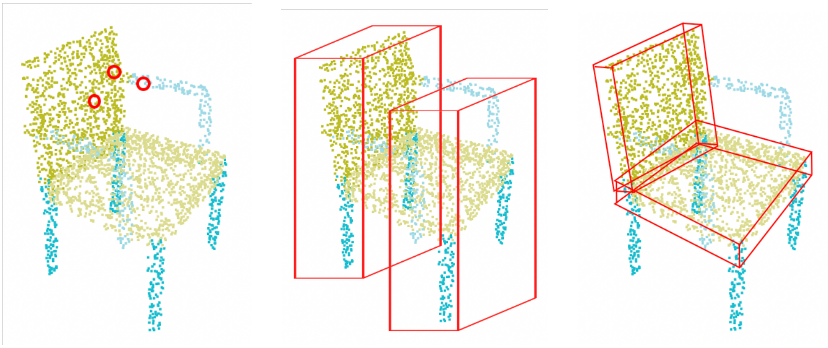
3.1 Attention on 2D Images

In this subsection, we review some recent papers that apply the attention-based methods on computer vision tasks. Unlike most other survey papers [15, 19] that categorize the methods based on tasks, since we are more interested in their actual application details, we categorize them based on the ways in which attentions are applied. Figure 3.1 illustrates three different attention types based on the different input entities. An interesting variant is the combination of CNNs and attention. Some models first use CNNs to get feature maps and then apply attention on learned feature maps. In this case, the feature maps can be regarded as latent representative images, but still are images.

Pixel-wise attention: iGPT [5] is a famous generative pre-trained Transformer model that learns directly on pixels. In iGPT, images are firstly downsampled and flattened into pixels, and then attention-based learning techniques that similar to GPT [1] are applied to explore autoregressive or BERT objectives. Given an input of a certain length of pixels, the model learns to predict the next pixel; or, given an input with some pixels masked out, the model learns to predict the masked pixels. iGPT did not use any CNN, but it is also possible to combine CNNs with the attention mechanism. For example, DETR [2] does attention on flattened feature maps for image detection tasks, with positional encoding added. While DERT considers the pixel-wise attention only over the channel dimension, DANet [12] proposes to use a symmetric branch to do attention over both the channel and the position (height and depth) dimensions.

Patch-wise attention: Transformers are large models. For a high resolution image, it is too computational expensive to use every pixel as an input entity. To decrease number of input entities, apart from doing CNNs ahead, patch-wise attention is also an option. Vision Transformer (ViT) [9] is the first work to showcase how attention operations can fully replace convolution operations in deep neural networks on large-scale computer vision datasets. They applied the original Transformer model on a sequence of image patches, which are vectorized and projected to a patch embedding using a linear layer. Position embedding is attached with it to encode location information. The Transformer model was pre-trained on a large image dataset and later fine-tuned for downstream recognition benchmarks. For patch-wise attention, it is also possible to combine it with CNNs. For example, after getting the feature maps, IPT [4] groups patches along the channel dimension for attention computation.

Object-wise attention: Different from above two types, this type of attention models does not directly learn from the original images. It is more of a second process in the whole learning pipeline and requires some additional information output from the previous process. For example, based on rough detection results, Relation Networks [17] processes a set of objects simultaneously through interaction between their appearance feature and geometry, thus allowing modeling of their relations. Moreover, when the time dimension is involved, doing object-wise attention also enhance the performance for tasks like action recognition and object tracking in videos. Interestingly, when input entities can be formalized as graphs, attentions can be applied not only in the fashion of key-query attention. For example, in the work of [27] which does action recognition on videos, graph-based attention is applied to learn object-wise relations. See more relevant discussions on graph-based attention in subsection 3.2.



(a) Point-wise self-attention (b) Patch-wise self-attention (c) Part-wise self-attention

Figure 3.2: Different attention types based on the different input entities from point clouds: (a) point-wise self-attention, attention is applied over all points or only neighbor points; (b) patch-wise self-attention, special subsample operations may be developed for patch choices; (c) part-wise self-attention, points of semantic parts or 3D bounding boxes may be used as input entities, in scene point clouds it would be object-wise attention.

3.2 Attention on 3D Data

Apart from RGBD images and multi-view images, other widely used 3D data representations are volumetric data, point clouds, meshes, etc. For voxel-based data representation, except for the methods on point clouds that voxelize the whole point cloud space [23, 29], there is really very little work that applies attention on volumetric data directly. We only find one work that does volumetric spatial and channel attention on medical images for segmentation and detection [28], but the method still treats input as image-based structure, other than in 3D. For mesh data representation, there are also only a few works that apply attention on them, e.g. METRO [22] uses Transformer models for human pose and mesh reconstruction. We believe there will be more explorations in applying attention on those data representations in the following years.

Of all the 3D data representations, point clouds are mostly investigated since they are more often used in the real-world applications, e.g. autonomous driving or industrial inspection. Hence, in the following part of this subsection, we mainly review the attention-based models for point clouds. Figure 3.2 illustrates three different attention types based on the different input entities from point clouds. Applying attention on point clouds is not the same as applying attention on images. Unlike images that are composed of well-aligned pixels, points in a point cloud are usually unordered and randomly positioned. This has pros and cons. On one side, it requires no positional encoding since we want to rule out the influence of point order. On the other side, having an unfixed number of neighbor points makes the common convolution operations unfeasible. In short, point clouds do not have image-like feature maps, thus the combination of convolution and attention operations becomes difficult.

Point-wise attention: PCT [14] pioneered on this topic by replacing the encoder in the original PointNet [24] framework with some attention-based layers. Skip links are also used for better latent feature acquisition. In [30], the point Transformer layer is sandwiched between two linear layers to create a point Transformer block, which is stacked multiple times in the proposed network architecture. Their design also includes transition down/up blocks to reduce/increase the number of points between consecutive layers, in a typical encoding-decoding style.

Different from above methods that use key-query attention for computing the attention score, there are also other methods use MLP and softmax layers directly to compute the attention score. RandLA-Net [18] learns attention scores for points as a soft mask to replace the original max/mean/sum pooling layer for better feature pooling. GAPNet [3] and Liang et al. [21] do similar point-wise self-attention with neighbor points to learn attention coefficients from them for each point. We term this type of attention as aforementioned graph-based attention, since it is usually applied on graph structure data (finding neighbor relations for each point is like building edges that defined in graphs).

Patch-wise attention: In order to perform parallel computation, if patches are not processed into fixed-length latent representation firstly, they need to be of the same size as input entities. Unlike images which are easy to cut into same size of patches, how to subsample the point clouds into patches is still an open question. Engel et al. [11] proposes an interesting idea of SortNet, with which sub-point clouds of same size can be learned. After that, the attention operation is applied on latent features of sub-point clouds and the global feature to perform local-global attention. However, the patches learned with SortNet are only of little semantic meanings. Even with multiple-SortNets, only a small ratio of points are selected to represent the input shape. Improvements may be made on subsample operations to get more representative patches.

Part-wise attention: This type of attention learns relations between semantic areas in a point cloud. For shape point clouds, it is part-wise self-attention. For scene point clouds, it is object-wise self-attention. Same as in applying object-wise attention on images, here it also requires a former step of getting some additional information. For example, MPAN [20] firstly does segmentation on the input shape point cloud, then do part-wise self-attention over segmented parts to get a shape descriptor for shape retrieval tasks.

4 Research Ideas for Attention with 3D Data

As discussed above, only a few researchers work on applying the attention mechanism directly on 3D Non-Euclidean data. Making some possible progresses on this topic would be exciting. In this section, two research ideas are proposed in this domain. One for voxel-based data, the other for point cloud data.

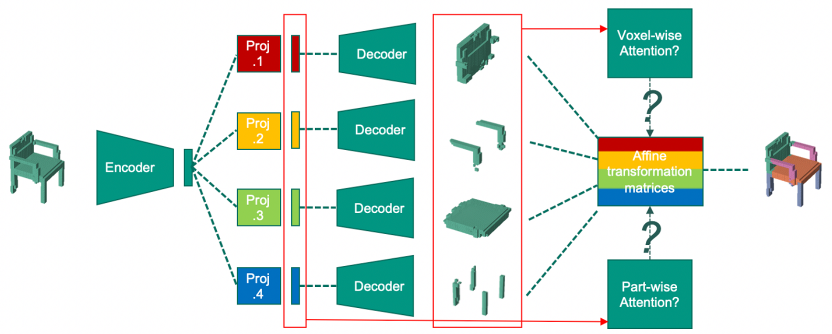


Figure 4.1: Proposed pipeline of the VoxAttention framework.

4.1 VoxAttention: Volumetric Shape Synthesis via Part Assembly

As discussed in section 3.2, apart from the methods that perform voxelization on point clouds, works that directly apply the attention mechanism on 3D volumetric data are really rare. We think it would be an interesting topic to research on. Among all the voxel data-based tasks, deep generative models-based 3D shape synthesis is getting more attention recently since they are not only good at data reconstruction, but also helpful in producing meaningful latent representations. Current existing 3D shape synthesis methods can be divided into two kinds: structure-oblivious ones and structure-aware ones. Compared to structure-oblivious methods, structure-aware methods introduce additional semantic information and thus result in better performances. They are starting to gain more and more attention in this domain. Schor et al. [25] train part-wise generators and a part composition network for the generation of 3D point clouds. Dubrovina et al. [10] propose a decomposer-composer framework to learn a factorized shape embedding space for part-based 3D volumetric shape modeling.

Inspired by the work of [10], we propose a new generator-assembler framework for 3D volumetric shape synthesis, with the additional help of self-attention mechanism. Figure 4.1 illustrates its basic idea. Firstly, the binary voxel grid is fed into an encoder, resulting in a latent encoding. Then several projection matrices are used to project it to different part latent representations, and a parameter-shared decoder is used to reconstruct the parts. To disentangle the semantic information as much as possible, the projection matrices are mutual orthogonal. Note the reconstructed parts now are enlarged and centered in a 32^3 voxel space. To learn respective affine transformation matrices for part assembly, instead of using a straightforward idea to apply CNNs on the reconstructed parts like [10], our proposal is to use the attention mechanism. For example, we can apply part-wise attention over part latent representations, or do voxel-wise attention on reconstructed parts to learn spatial relations between them. With our proposed attention-based method, ideally, the network should be able to learn dynamic transformation matrices for part latent representations. Even if some part latent representations are swapped, the network should still be able to reconstruct shapes with relatively correct part size and position.

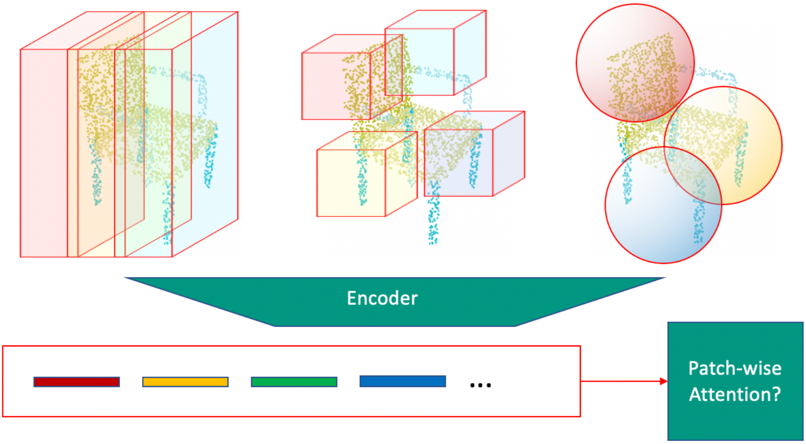


Figure 4.2: Proposed pipeline of the PointAttention framework.

4.2 PointAttention: Patch-wise Attention-based Learning on Point Clouds

Both key-query attention and graph-based attention are more often used for the point cloud-related tasks. However, most of these works only focus on point-wise self-attention. We propose to explore more on patch-wise self-attention. As discussed in subsection 3.2, how to subsample the point clouds into patches is still an open question. Unlike [23] or [29] in which the point cloud space get voxelized, we directly use pre-designed subsampling methods to cut out patches, as illustrated on the top of Figure 4.2.

A key point is to normalize either the input or the latent feature to the same size. In [23, 29], although each voxel contains different number of points, voxels' feature maps or latent representations are of same size. In our case, we can directly subsample the original point clouds into same size of sub-point clouds and use them as the network input instead. Apart from the direct FPS (short for farthest point sampling) subsample operation, we propose such a new one: firstly slice the point cloud or use a cuboid/sphere to cut the point cloud, then apply FPS on these sub-point clouds to get patches of a required point number. Afterwards, patch-wise attention may be applied on latent representations encoded from different patches to learn patch relations for better point cloud classification and segmentation. Another idea is that instead of pre-defining the subsample operations manually, we can define a module similar to SortNet [11] to let it learn better patches by itself automatically.

If the original point clouds are not of large size, the subsamples can still somehow represent the 3D shape partially with semantic meanings. In this case, the subsampling operations are augmentation-like operations, hence the subsamples are also ideal input data for contrastive learning [6, 13]. Combining the attention mechanism and contrastive learning, other new self-supervised learning methods may also be proposed.

5 Conclusion

Attention mechanism plays an important role in learning feature relations between input entities, especially for the long-range relations. In this report, we review a variety of attention-based approaches that are used in the NLP or the CV domain. In the CV domain, based on the type of the input entities, we categorize the attention-based methods into different types for 2D data and 3D data respectively. We also share some insights from this perspective. Finally, since applying attention on 3D data is relatively more difficult and there exists only a few works, we propose two possible research ideas in this scope. Future researches will firstly take the proposed ideas into consideration and perform experiments on them. We also hope this technical report may inspire other researchers that are interested in this topic.

References

[1] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: *ArXiv* abs/2005.14165 (2020).

[2] Nicolas Carion et al. “End-to-End Object Detection with Transformers”. In: *ArXiv* abs/2005.12872 (2020).

[3] Can Chen, Luca Zanotti Fragonara, and Antonios Tsourdos. “GAPNet: Graph Attention based Point Neural Network for Exploiting Local Feature of Point Cloud”. In: *Neurocomputing* 438 (2021), pp. 122–132.

[4] Hanting Chen et al. “Pre-Trained Image Processing Transformer”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 12294–12305.

[5] Mark Chen et al. “Generative Pretraining From Pixels”. In: *ICML*. 2020.

[6] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *ArXiv* abs/2002.05709 (2020).

[7] Junyoung Chung et al. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: *ArXiv* abs/1412.3555 (2014).

[8] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *NAACL*. 2019.

[9] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *ArXiv* abs/2010.11929 (2021).

[10] Anastasia Dubrovina et al. “Composite Shape Modeling via Latent Space Factorization”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 8139–8148.

[11] Nico Engel, Vasileios Belagiannis, and Klaus C. J. Dietmayer. “Point Transformer”. In: *IEEE Access* 9 (2021), pp. 134826–134840.

[12] J. Fu et al. “Dual Attention Network for Scene Segmentation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 3141–3149.

[13] Jean-Bastien Grill et al. “Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning”. In: *ArXiv* abs/2006.07733 (2020).

[14] Meng-Hao Guo et al. “PCT: Point Cloud Transformer”. In: *Comput. Vis. Media* 7 (2021), pp. 187–199.

[15] Kai Han et al. “A Survey on Visual Transformer”. In: *ArXiv* abs/2012.12556 (2020).

[16] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9 (1997), pp. 1735–1780.

[17] Han Hu et al. “Relation Networks for Object Detection”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 3588–3597.

[18] Qingyong Hu et al. “RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 11105–11114.

[19] Salman Hameed Khan et al. “Transformers in Vision: A Survey”. In: *ArXiv* abs/2101.01169 (2021).

[20] Ziru Li et al. “MPAN: Multi-Part Attention Network for Point Cloud Based 3D Shape Retrieval”. In: *IEEE Access* 8 (2020), pp. 157322–157332.

[21] Zhidong Liang et al. “3D Instance Embedding Learning With a Structure-Aware Loss Function for Point Cloud Segmentation”. In: *IEEE Robotics and Automation Letters* 5 (2020), pp. 4915–4922.

[22] Kevin Lin, Lijuan Wang, and Zicheng Liu. “End-to-End Human Pose and Mesh Reconstruction with Transformers”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 1954–1963.

[23] Jiageng Mao et al. “Voxel Transformer for 3D Object Detection”. In: *ArXiv* abs/2109.02497 (2021).

[24] C. Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 77–85.

[25] Nadav Schor et al. “CompoNet: Learning to Generate the Unseen by Part Synthesis and Composition”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 8758–8767.

[26] Ashish Vaswani et al. “Attention is All you Need”. In: *ArXiv* abs/1706.03762 (2017).

[27] X. Wang and Abhinav Gupta. “Videos as Space-Time Region Graphs”. In: *ECCV*. 2018.

[28] Xudong Wang et al. “Volumetric Attention for 3D Medical Image Segmentation and Detection”. In: *MICCAI*. 2019.

[29] Cheng Zhang et al. “PVT: Point-Voxel Transformer for 3D Deep Learning”. In: 2021.

[30] Hengshuang Zhao et al. “Point Transformer”. In: *ArXiv* abs/2012.09164 (2020).