A Survey of Visual Transformers

Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He

***Abstract*— Transformer, an attention-based encoder–decoder model, has already revolutionized the field of natural language processing (NLP). Inspired by such significant achievements, some pioneering works have recently been done on employing Transformer-liked architectures in the computer vision (CV) field, which have demonstrated their effectiveness on three fundamental CV tasks (classification, detection, and segmentation) as well as multiple sensory data stream (images, point clouds, and vision-language data). Because of their competitive modeling capabilities, the visual Transformers have achieved impressive performance improvements over multiple benchmarks as compared with modern convolution neural networks (CNNs). In this survey, we have reviewed over 100 of different visual Transformers comprehensively according to three fundamental CV tasks and different data stream types, where taxonomy is proposed to organize the representative methods according to their motivations, structures, and application scenarios. Because of their differences on training settings and dedicated vision tasks, we have also evaluated and compared all these existing visual Transformers under different configurations. Furthermore, we have revealed a series of essential but unexploited aspects that may empower such visual Transformers to stand out from numerous architectures, e.g., slack high-level semantic embeddings to bridge the gap between the visual Transformers and the sequential ones. Finally, two promising research directions are suggested for future investment. We will continue to update the latest articles and their released source codes at `https://github.com/liuyang-ict/awesome-visual-transformers`.**

***Index Terms*— Classification, computer vision (CV), detection, point clouds, segmentation, self-supervision, visual-linguistic pretraining, visual Transformer.**

I. Introduction

TRANSFORMER [1], which adopts an attention-based structure, has first demonstrated its tremendous effects on the tasks of sequence modeling and machine translation. As shown in Fig. 1, Transformers have gradually emerged as the predominant deep learning models for many natural language processing (NLP) tasks. The most recent dominant models are the self-supervised Transformers, which are pretrained over sufficient datasets and then fine-tuned over a small sample set for a given downstream task [2], [3], [4], [5], [6], [7], [8], [9]. The generative pretrained transformer (GPT) families [2], [3], [4] leverage the Transformer decoders to enable autoregressive language modeling, while the bidirectional encoder representations from transformers (BERT) [5] and its variants [6], [7] serve as autoencoder language models built on the Transformer encoders.

In the computer vision (CV) field, prior to the visual Transformers, convolution neural networks (CNNs) have emerged as a dominant paradigm [10], [11], [12]. Inspired by the great success of such self-attention mechanisms (Fig. 2) for the NLP tasks [1], [13], some CNN-based models attempted to capture the long-range dependencies through adding a self-attention layer at either spatial level [14], [15], [16] or channel level [17], [18], [19], while others try to replace the traditional convolutions entirely with the global [20] or local self-attention blocks [21], [22], [23], [24], [25], [26], [27]. Although Ramachandran et al. [24] have demonstrated the efficiency of self-attention block without the help from CNNs, such pure attention model is still inferior to the state-of-the-art (SoTA) CNN models on the prevailing benchmarks.

With the grateful achievements of linguistic Transformers and the rapid development of visual attention-based models, numerous recent works have migrated the Transformers to the CV tasks, and some comparable results have been achieved. Cordonnier et al. [28] theoretically demonstrated the equivalence between multihead self-attention (MHSA) and CNNs, and they designed a pure Transformer by using patch downsampling and quadratic position encoding to verify their theoretical conclusion. Dosovitskiy et al. [29] further extended such a pure Transformer for large-scale pretraining, which has achieved SoTA performance over many benchmarks. In addition, the visual Transformers have also obtained great performances for other CV tasks, such as detection [30], segmentation [31], [32], tracking [33], and generation [34].

As shown in Fig. 1, following the pioneer works [29], [30], hundreds of Transformer-based models have been proposed for various vision applications within the last year. Thus, a systematic literature survey is strongly desired to identify, categorize, and evaluate these existing visual Transformers. Considering that the readers may come from different areas, we review all these visual Transformers according to three fundamental CV tasks (i.e., classification, detection, and segmentation) and different types of data streams (i.e., images, point clouds, and multistream data). As shown in Fig. 3, this survey categorizes

Manuscript received 10 November 2021; revised 29 April 2022 and 24 July 2022; accepted 26 November 2022. *(Corresponding authors: Zhiqiang He; Zhongchao Shi; Yang Zhang.)*

Yang Liu, Yao Zhang, and Feng Hou are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100000, China, and also with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100000, China (e-mail: liuyang20c@mails.ucas.ac.cn).

Yixin Wang is with the School of Engineering, Stanford University, Palo Alto, 94305 USA.

Jin Yuan is with the School of Computer Science and Engineering, Southeast University, Nanjing 214135, China.

Jiang Tian, Yang Zhang, Zhongchao Shi, and Jianping Fan are with the AI Lab, Lenovo Research, Beijing 100000, China (e-mail: shizc2@lenovo.com; zhangyang20@lenovo.com).

Zhiqiang He is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100000, China, also with the University of Chinese Academy of Sciences, Beijing 100000, China, and also with Lenovo Ltd., Beijing 100000, China (e-mail: hezq@lenovo.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2022.3227717.

Digital Object Identifier 10.1109/TNNLS.2022.3227717

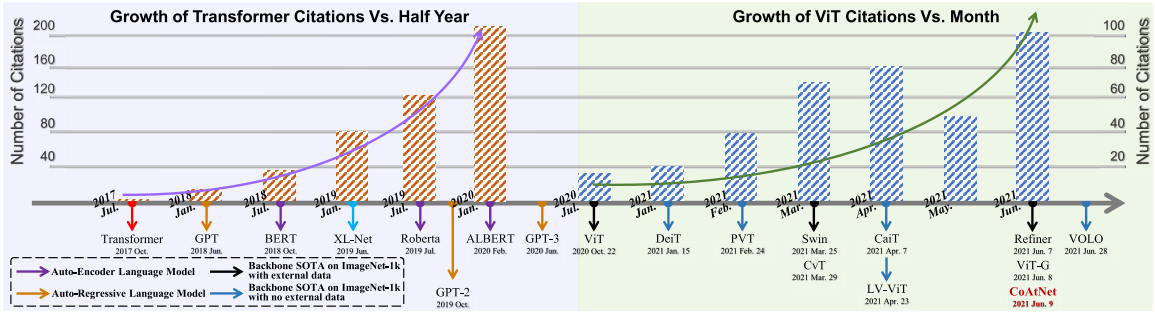

Fig. 1. Odyssey of Transformer application and growth curves of both Transformer [1] and ViT [29] citations according to Google Scholar. Top left: growth of Transformer citations in the top linguistics and machine learning conference publications. Top right: growth of ViT citations in Arxiv publications. Bottom left: Odyssey of language model [1], [2], [3], [4], [5], [6], [7], [8]. Bottom right: Odyssey of visual Transformer backbone where the black [29], [35], [36], [37], [38], [39] is the SoTA with external data and the blue [40], [41], [42], [43], [44] refers to the SoTA without external data (best viewed in color).

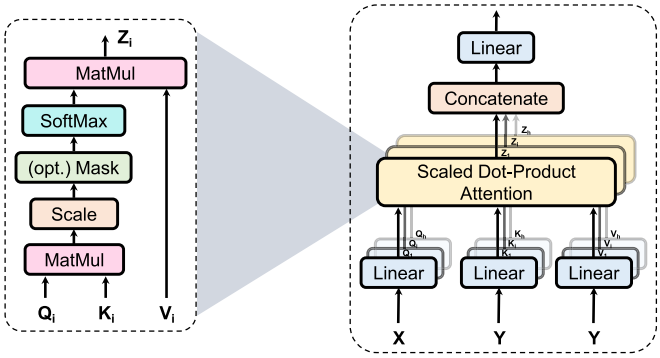

Fig. 2. Structure of the attention layer. Left: scaled dot-product attention. Right: multihead attention mechanism.

all these existing methods into multiple groups according to their dedicated vision tasks, data stream types, motivations, and structural characteristics.

Before us, several reviews have been published. Tay et al. [45] reviewed the efficiency of the linguistic Transformers, Khan et al. [46] and Han et al. [47] summarized the early visual Transformers and attention-based models, and Lin et al. [48] provided a systematic review of various linguistic Transformers and sketchy vision applications. Distinctively, this article provides a more comprehensive review of the most recent visual Transformers and categorizes them systematically.

(1) *Comprehensiveness and Readability.* This article comprehensively reviews over 100 visual Transformers according to their applications on three fundamental CV tasks (i.e., classification, detection, and segmentation) and different types of data streams (i.e., image, point clouds, and multistream data). We select more representative methods with detailed descriptions and analyses but introduce other related works briefly. In addition to analyzing each model independently, we also build their internal connections from certain senses such as progressive, contrastive, and multiview analysis.

(2) *Intuitive Comparison.* As these visual Transformers follow different training schemes and hyperparameter settings for various vision tasks, this survey presents multiple lateral comparisons over different datasets and restrictions. More importantly, we summarize a series of promising components designed for each task, including shallow local convolution with hierarchical structure for backbone, spatial prior acceleration with sparse attention for neck detector, and general-purpose mask prediction scheme for segmentation.

(3) *In-depth Analysis.* We further provide well-thought insights from the following aspects.
   a) How visual Transformers bridge the traditional sequential tasks to the visual ones (why does Transformer work effectively in CV).
   b) The correspondence between the visual Transformers and other neural networks.
   c) The double edges of the visual Transformers.
   d) The correlation of the learnable embeddings (i.e., class token, object query, and mask embedding) adopted in different tasks and data stream types. Finally, we outline some future research directions. For example, the encoder–decoder Transformer backbone can unify multiple visual tasks and data stream types through query embeddings.

The rest of this article is organized as follows. An overview of the architectures and the critical components for the vanilla sequential Transformers are introduced in Section II. A comprehensive taxonomy for the Transformer backbones is summarized in Section III with a brief discussion of their applications for image classification. We then review contemporary Transformer detectors, including Transformer necks and backbones in Section IV. Section V clarifies the mainstream and its variants for the visual Transformers in the segmentation field according to their embedding forms (i.e., patch embedding and query embedding). Sections III–V also briefly analyze a specific aspect of their corresponding fields with performance evaluation. In addition to 2-D visual recognition, Section VI briefly introduces the recently developed 3-D visual recognition from the perspective of point clouds. Section VII further overviews the fusion approaches within the visual Transformers for multiple data stream types (e.g., multiview, multimodality, visual-linguistic pretraining, and visual grounding). Finally, Section VIII provides three aspects for further discussion and points out some promising research directions for future investment.

II. Original Transformer

The original Transformer [1] is first applied to the task for sequence-to-sequence autoregression. Compared with previous sequence transduction models [49], [50], such original Transformer inherits the encoder–decoder structure but discards the recurrence and convolutions entirely by using multihead

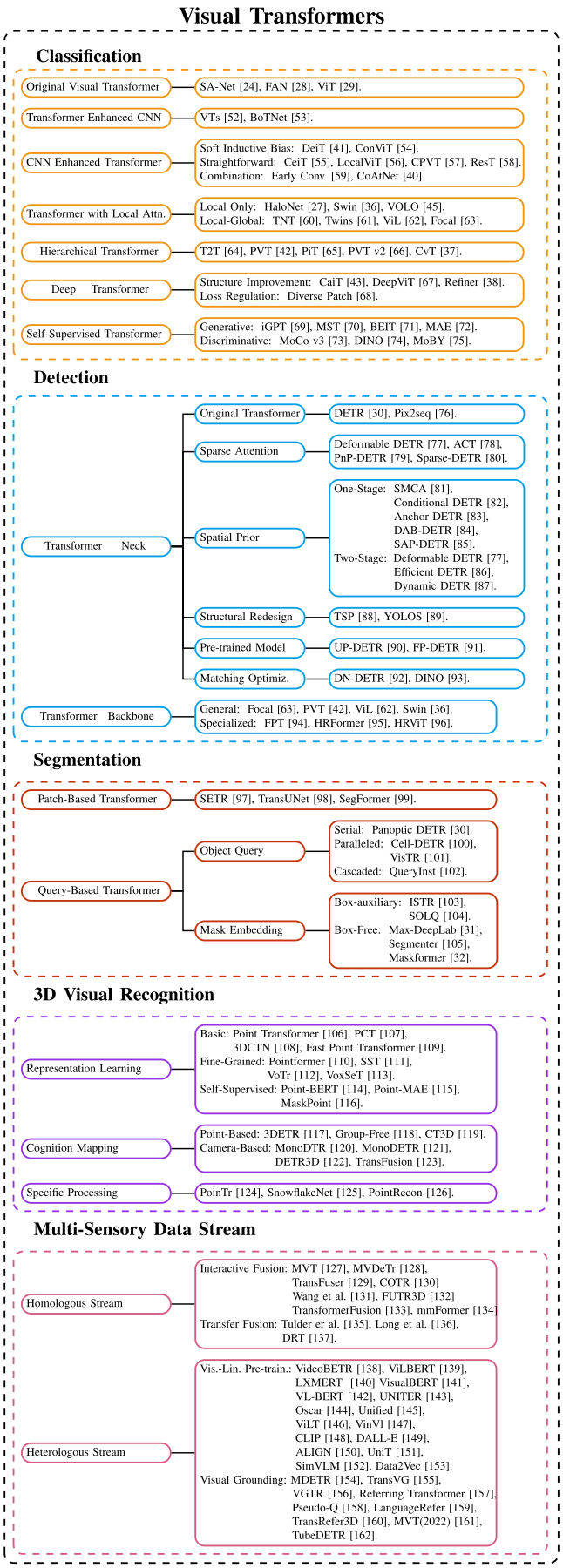

Fig. 3. Taxonomy of visual Transformers (best viewed in color).

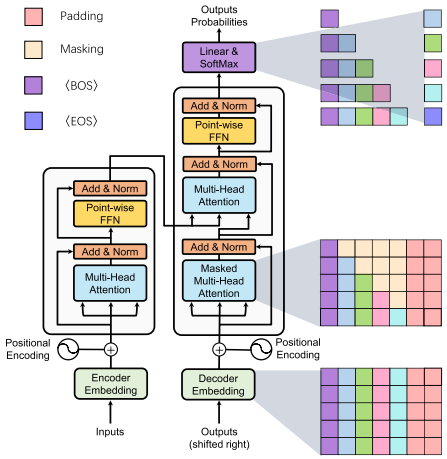

Fig. 4. Overall architecture of Transformer [1]. The 2-D lattice represents each state of queries during training (best viewed in color).

attention mechanisms and pointwise feedforward networks (FFNs). In the following, we will provide an architectural overview of the original Transformer.

A. Multihead Attention Mechanism

The mechanism with one single head attention can be grouped into two parts: 1) a transformation layer maps the input sequences $X \in \mathbb{R}^{n_x \times d_x}$ and $Y \in \mathbb{R}^{n_y \times d_y}$ into three different vectors (query Q, key K, and value V), where n and d are the length and the dimension of the inputs, respectively, and 2) an attention layer, as shown in Fig. 2, explicitly aggregates the query with the corresponding key, assigns them to the value, and updates the output vector.

The formula for the transformation layer is defined as

$$Q = XW^Q, \quad K = YW^K, \quad V = YW^V \tag{1}$$

where $W^Q \in \mathbb{R}^{d_x \times d^k}$, $W^K \in \mathbb{R}^{d_y \times d^k}$, and $W^V \in \mathbb{R}^{d_y \times d^v}$ are linear matrices; and d^k and d^v are the dimension of the query–key pair and the value that are projected from Y and X, respectively. Such two sequence inputs are referred to as the cross-attention mechanism. It can also be regarded as self-attention when $Y = X$. In form, self-attention is applied to both Transformer encoder and decoder, while the cross attention severs as a junction within the decoder.

Then, the scale-dot attention mechanism is formulated as

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where the attention weights are generated by a dot-product operation between Q and K, and a scaling factor $(d_k)^{1/2}$ and a softmax operation are supplied to translate the attention weights into a normalized distribution. The resulting weights are assigned to the corresponding value elements, thereby yielding the final output vector.

Due to the restricted feature subspace, the modeling capability of the single-head attention block is quite coarse. To tackle this issue, as shown in Fig. 2, an MHSA mechanism is proposed to linearly project the input into multiple feature subspaces and process them by using several independent attention heads (layers) parallelly. The resulting vectors are

concatenated and mapped to the final outputs. The process of MHSA can be formulated as

$$\begin{aligned} Q_i &= XW^{Q_i}, \quad K_i = XW^{K_i}, \quad V_i = XW^{V_i} \\ Z_i &= \text{Attention}(Q_i, K_i, V_i), \quad i = 1 \ldots h \\ \text{MultiHead}(Q, K, V) &= \text{Concat}(Z_1, Z_2, \ldots, Z_h)W^O \end{aligned} \tag{3}$$

where h is the head number, $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ denotes the output projected matrix, Z_i denotes the output vector of each head, and $W^{Q_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W^{K_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $W^{V_i} \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are three different groups of matrices. Multihead attention separates the inputs into h independent attention heads with d_{model}/h-dimensional vectors and integrates each head feature dependently. Without extra costs, multihead attention enriches the diversity of the feature subspaces.

B. Positionwise FFNs

The output of MHSA is then fed into two successive FFNs with a ReLU activation as

$$\text{FFN}(x) = \text{RELU}(W_1 x + b_1)W_2 + b_2. \tag{4}$$

This positionwise feedforward layer can be viewed as a pointwise convolution, which treats each position equally but uses different parameters between each layer.

C. Positional Encoding

Since the Transformer/attention operates on the input embedding simultaneously and identically, the order of the sequence is neglected. To make use of the sequential information, a common solution is to append an extra positional vector to the inputs, hence term the "positional encoding." There are many choices for positional encoding. For example, a typical choice is cosine functions with different frequencies as

$$\begin{aligned} \text{PE}_{(\text{pos},i)} &= \begin{cases} \sin(\text{pos} \cdot \omega_k), & \text{if} \quad i = 2k \\ \cos(\text{pos} \cdot \omega_k), & \text{if} \quad i = 2k+1 \end{cases} \\ \omega_k &= \frac{1}{10\,000^{2k/d}}, \quad k = 1, \ldots, d/2 \end{aligned} \tag{5}$$

where pos and d are the position and the length of the vector, respectively, and i is the index of each element within vector.

D. Transformer Model

Fig. 4 shows the overall Transformer models with the encoder–decoder architecture. Specifically, it consists of N successive encoder blocks, each of which is composed of two sublayers: 1) an MHSA layer aggregates the relationship within the encoder embeddings and 2) a positionwise FFN layer extracts feature representations. For the decoder, it also involves N consecutive blocks that follow a stack of the encoders. Compared with the encoder, each decoder block appends to a multihead cross-attention layer to aggregate both decoder embeddings and encoder outputs, where Y corresponds to the former and X is the latter as shown in (1). Moreover, all of the sublayers in both encoder and decoder employ a residual connection [11] and a layer normalization [162] to enhance the scalability of the Transformer. In order to record the sequential information, each input embedding is attached with a positional encoding at the beginning of the encoder stack and the decoder stack. Finally, a softmax operation is used for predicting the next word.

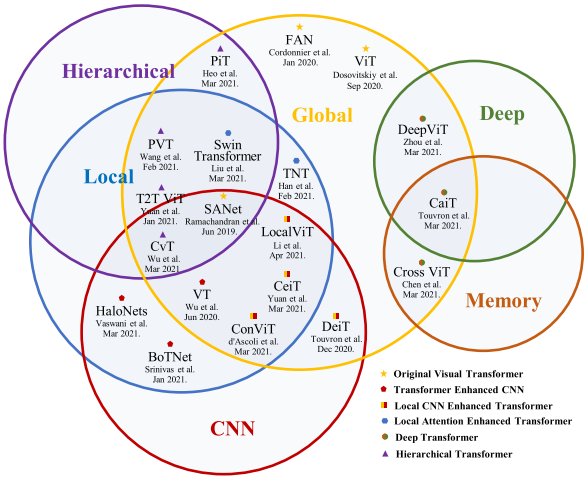

Fig. 5. Taxonomy of visual Transformer backbone (best viewed in color).

In an autoregressive language model, the Transformer is originated from the machine translation tasks. Given a sequence of words, the Transformer vectorizes the input sequence into the word embeddings, adds the positional encodings, and feeds the resulting sequence of the vectors into an encoder. During training, as shown in Fig. 4, Vaswani et al. [1] designed a masking operation according to the rule for the autoregressive task, where the current position only depends on the outputs of the previous positions. Based on this masking, the Transformer decoder is able to process the sequence of the input labels parallelly. During the inference time, the sequence of the previously predicted words is processed by the same operation to predict the next word.

III. Transformer for Classification

Following the prominent developments of the Transformers in NLP recent works attempt to introduce visual Transformers for image classification. This section comprehensively reviews over 40 visual Transformers and groups them into six categories, as shown in Fig. 5. We start with introducing the fully attentional network [24], [28] and the vision Transformer (ViT) [29] that first demonstrates Transformer efficacy on large scale classification benchmarks. Then, we discuss Transformer-enhanced CNN methods that utilize Transformer to enhance the representation learning in CNNs. Due to the negligence of local information in the original ViT, the CNN-enhanced transformer employs an appropriate convolutional inductive bias to augment the visual Transformer, while the local attention-enhanced Transformer redesigns patch partition and attention blocks to improve their locality. Following the hierarchical and deep structures in CNNs [163], the hierarchical Transformer replaces the fixed-resolution columnar structure with a pyramid stem, while the deep Transformer prevents the attention map from oversmooth and increases its diversity in the deep layer. Moreover, we also review the existing visual Transformers with self-supervised learning. Finally, we make a brief discussion based on intuitive comparisons for further investigation. More visual Transformers' milestones are introduced in the Supplementary Material.

A. Original Visual Transformer

Inspired by the tremendous achievements of the Transformers in the NLP field [2], [3], [4], [5], the previous technology trends for the vision tasks [14], [15], [16], [17], [164] incorporate the attention mechanisms with the convolution models to augment the models' receptive field and global dependency.

Beyond such hybrid models, Ramachandran et al. [24] contemplated whether the attention can completely replace the convolution and then presented a stand-alone self-attention network (SANet), which has achieved superior performance on the vision tasks compared with the original baseline. Given a ResNet [11] architecture, the authors straightforwardly replace the spatial convolution layer (3×3 kernel) in each bottleneck block with a locally spatial self-attention layer and keep other structures the same as the original setting in ResNet. Moreover, lots of ablations have shown that the positional encodings and convolutional stem can further improve the network efficacy.

Following [24], Cordonnier et al. [28] pioneered a prototype design (called fully attentional network in their original paper), including a fully vanilla Transformer and quadratic positional encoding. They also theoretically proved that a convolutional layer can be approximated by a single MHSA layer with relative positional encoding and sufficient heads. With the ablations on CIFAR-10 [165], they further verify that such a prototype design does learn to attend a grid-like pattern around each query pixel, as their theoretical conclusion.

Different from [28] that only focuses on lite scale model, the ViT [29] further explores the effectiveness of the vanilla Transformer with large-scale pretrained learning, and such a pioneer work impacts the community significantly. Because the vanilla Transformers only accept the sequential inputs, the input image in ViT is first split into a series of nonoverlapped patches and they are then projected into patch embeddings. Then, a 1-D learnable positional encoding is added on the patch embeddings to retain the spatial information, and the joint embeddings are then fed into the encoder, as shown in Fig. 6. Similar to BERT [5], a learned [class] token is attached with the patch embeddings to aggregate the global representation and it serves as the final output for classification. Moreover, a 2-D interpolation complements the pretrained positional encoding to maintain the consistent order of the patches when the feeding images are in arbitrary resolution. By pretraining with a large-scale private dataset (JFT-300M [166]), ViT has achieved similar or even superior results on multiple image recognition benchmarks (ImageNet [167] and CIFAR-100 [165]) compared with the most prevailing CNNs methods. However, its generalization capability tends to be eroded with limited training data.

B. Transformer-Enhanced CNNs

As described in Section II, the Transformer has two keys: MHSA and FFN. There exists an approximation between the convolutional layer and the MHSA [28], and Dong et al. [168] suggested that the Transformer can further mitigate the strong bias of MHSA with the help of skip connections and FFN. Recently, some methods attempt to integrate the Transformer into CNNs to enhance representation learning. VTs [51] decouple semantic concepts for the input image into different channels and relate them densely through the encoder block,

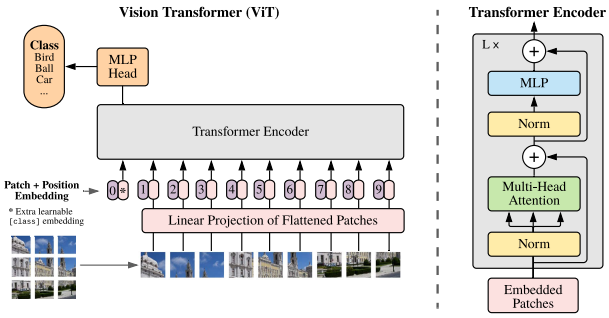

Fig. 6. Illustration of ViT. The flatten image patches with an additional class token are fed into the vanilla Transformer encoder after positional encoding. Only the class token can be predicted for classification (from [29]).

namely, VT-block. Such VT-block substitutes the last convolution stage to enhance the CNN model's ability on semantic modeling. Unlike previous approaches that directly replace convolution with attention structure, Srinivas et al. [52] proposed a conceptual redefinition that the successive bottleneck blocks with MHSA can be formulated as the bottleneck Transformer (BoTNet) blocks. The relative position encoding [169] is adopted to further mimic the original Transformer. Based on ResNet [11], BoTNet outperforms the most CNN models with similar parameter settings on the ImageNet benchmark and further demonstrates the efficacy of hybrid models.

C. CNN-Enhanced Transformer

Inductive bias is defined as a set of assumptions on data distribution and solution space, whose manifestations within convolution are the locality and the translation invariance [170]. As the covariance within local neighborhoods is large and tends to be gradually stationary across an image, CNNs can process an image effectively with the help of the biases. Nevertheless, strong biases also limit the upper bound of CNNs when sufficient data are available. Recent efforts attempt to leverage an appropriate CNN bias to enhance Transformer.

Touvron et al. [40] proposed a data-efficient image Transformer (DeiT) to moderate the ViT's dependence on large datasets. In addition to the existing strategies for data augmentation and regularization, a teacher–student distillation strategy is applied for auxiliary representation learning, where the student ViT is attached with a distilled token supervised by the pseudo label from the teacher model. Extensive experiments have demonstrated that CNN is a better teacher model than the Transformer. Surprisingly, the distilled student Transformer even outperforms its teacher CNN model. These observations are explained in [171]: the teacher CNN transfers its inductive bias in a soft way to the student Transformer through knowledge distillation. Based on ViT's architecture, DeiT-B attains the top-1 accuracy of 85.2% without external data. ConViT [53] appends a parallel convolution branch with vanilla Transformer to impose inductive biases softly. The main idea of the convolution branch is a learnable embedding that is first initialized to approximate the locality as similar to the convolution and then explicitly gives each attention head freedom to escape the locality by adjusting a learned gating parameter. CeiT [54] and LocalViT [55] extract the locality by directly adding a depthwise convolution in FFN. As pointwise convolution is equal to positionwise FFN, they

extend FFN to an inverted residual block [172] to build a depthwise convolutional framework. Based on the assumption of positional encoding [57] and the observation in [173], ResT [57] and CPVT [56] try to adapt the inherent positional information of the convolution to the arbitrary size of inputs instead of interpolating the positional encoding. Including CvT [36], these methods replace the linear patch projection and positional encoding with the convolution stacks. Both methods benefit from such convolutional position embedding, especially for small model.

Besides the "internal" fusion, many approaches focus on "apparent" combinations according to different visual Transformer's structures. For standard columnar structure, Xiao et al. [58] substituted the original patchify stem (single non-overlap large kernel) with several stacked stride-2 3 × 3 kernels. Such a convolutional stem significantly improves ViT by 1%–2% accuracy on ImageNet-1k and facilitates its stability and generalization on the downstream tasks. For hierarchical structures, Dai et al. [39] investigated an optimal combination of hybrid models to benefit the performance tradeoff. By comparing a series of hybrid models, they propose a convolution and attention network (CoAtNet) to leverage the strength of both CNNs and Transformer. They observe that using convolution in the early stages is more effective than transformers, and depthwise convolution can be naturally integrated into attention blocks for hierarchical structures. It has achieved the SoTA performance across multiple datasets.

D. Local Attention-Enhanced Transformer

The coarse patchify process in ViT [29] neglects the local image information. In addition to adding CNNs, various local attention mechanisms are proposed to dynamically attend the neighbor elements and augment the local extraction ability.

One of the representative methods is the shifted windows (Swin) Transformer [35]. Similar to TSM [174] [Fig. 7(a)], Swin utilizes a shifted window along the spatial dimension to model the global and boundary features. In detail, two successive windowwise attention layers can facilitate the cross-window interactions [Fig. 7(b)–(c)], similar to the receptive field expansion in CNNs. Such operation also reduces the computational complexity from $O(2n^2C)$ to $O(4M^2nC)$ in one attention layer, where n and M denote the patch length and the window size, respectively. Swin Transformer achieves the 84.2% accuracy on ImageNet and the latest SoTA on multiple dense prediction benchmarks (see Section IV-B).

Inspired by [175], Han et al. [59] leveraged a Transformer-iN-Transformer (TNT) model to aggregate both patch- and pixel-level representations. Each layer of TNT consists of two successive blocks: an inner block models the pixelwise interaction within each patch and an outer block extracts the global information. Twins [60] employs a spatially separable self-attention mechanism, similar to depthwise convolution [172] or windowwise TNT [59], to model the local–global representation. Another separate form is ViL [61], which replaces the single class token with a series of local embeddings (termed as global memory). These local embeddings only perform inner attention and interaction with their corresponding 2-D spatial neighbors. VOLO [44] proposes outlook attention, which is similar to a patchwise dynamic convolution, to focus on the finer level features, including three operations: unfold, linear-weights attention, and refold. Based on [43], it achieves SoTA results on ImageNet without external data.

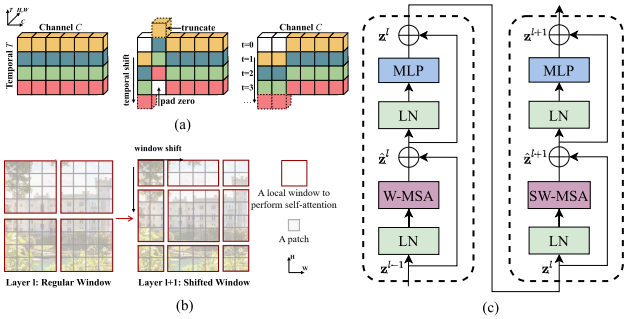

Fig. 7. Overview of Swin Transformer and TSM. (a) TSM with bidirectional and unidirectional operation. (b) Shifted window method. (c) Two successive Transformer blocks of Swin Transformer. The regular and shifted windows correspond to W-MSA and SW-MSA, respectively (from [35] and [174]).

E. Hierarchical Transformer

As its columnar structure produces a fixed resolution features across all Transformer layers, ViT [29] sacrifices fine-grained feature extraction and incurs substantial computational costs. Followed by the hierarchical models, Tokens-to-Token ViT (T2T-ViT) first introduces a paradigm of hierarchical Transformer and employs an overlapping unfold operation for downsampling. However, such an operation brings heavy memory and computation costs. Therefore, pyramid vision Transformer (PVT) [41] leverages a nonoverlapping patch partition to reduce the feature size. Furthermore, a spatial-reduction attention (SRA) layer is applied in PVT to further reduce the computational cost by learning low-resolution key–value pairs. Empirically, PVT adapts the Transformer to the dense prediction tasks on many benchmarks that demand large inputs and fine-grained features with computational efficiency. Moreover, both PiT [64] and CvT [36] utilize pooling and convolution to perform token downsampling, respectively. In detail, CvT [36] improves the SRA of PVT [41] by replacing the linear layer with a convolutional projection. Based on the convolutional bias, CvT [36] can adapt to arbitrary size inputs without positional encodings.

F. Deep Transformer

Empirically, increasing model's depth always strengthens its learning capacity [11]. Recent works apply a deep structure to Transformer and massive experiments are conducted to investigate its scalability by analyzing cross-patch [67] and cross-layer [37], [66] similarities, and the contribution of residual blocks [42]. In the deep Transformer, the features from the deeper layers tend to be less representative (attention collapse [66]), and the patches are mapped into the indistinguishable latent representations (patch oversmoothing [67]). To address such limitations mentioned above, current methods present the corresponding solutions from two aspects.

From the aspect of model's structure, Touvron et al. [42] presented efficient class attention in image Transformers (CaiT), including two stages: 1) multiple self-attention stages without class token, in each layer, a learned diagonal matrix initialized by small values is exploited to update the channel weights dynamically, thereby offering a certain degree of

TABLE I

TOP-1 ACCURACY COMPARISON OF VISUAL TRANSFORMERS ON IMAGENET-1K. "1K ONLY" DENOTES TRAINING ON IMAGENET-1K ONLY. "21K PRE." DENOTES PRETRAINING ON IMAGENET-21K AND FINE-TUNING ON IMAGENET-1K. "DISTILL." DENOTES APPLYING DISTILLATION TRAINING SCHEME OF DEIT [40]. THE COLOR OF "LEGEND" CORRESPONDING TO EACH MODEL ALSO DENOTES THE SAME MODEL IN FIG. 8

Method	#Params. (M)	FLOPs (G)	Acc. 1K	Acc. 21K/Distill.
ViT-B/16↑384 [29]	86.8	49.4	77.9	83.97†
ViT-L/16↑384 [29]	304.7	174.8	76.5	85.15†
VT-ResNet8 [52]	11.7	1.57	76.8	-
VT-ResNet34 [52]	19.2	3.24	79.9	-
VT-ResNet50 [52]	21.4	3.41	80.6	-
VT-ResNet101 [52]	41.5	7.13	82.3	-
BoTNet-T2 [53]	33.5	7.3	81.7	-
BoTNet-T4 [53]	54.7	10.9	82.8	-
BoTNet-T5↑256 [53]	75.1	19.3	83.5	-
DeiT-Ti [41]	5.7	1.1	72.2	74.5Ŧ
DeiT-S [41]	22.1	4.3	79.8	81.2Ŧ
DeiT-B [41]	86.6	16.9	81.8	83.4Ŧ
DeiT-B↑384 [41]	86.8	49.4	83.1	84.5Ŧ
ConViT-Ti [54]	6	1	73.1	-
ConViT-S [54]	27	5.4	81.3	-
ConViT-B [54]	86	17	82.4	-
CeiT-T [55]	6.4	1.2	76.4	-
CeiT-S [55]	24.2	4.5	82.0	-
CeiT-T↑384 [55]	6.4	3.6	78.8	-
CeiT-S↑384 [55]	24.2	12.9	83.3	-
LocalViT-T [56]	5.9	1.3	74.8	-
LocalViT-S [56]	22.4	4.6	80.8	-
ResT-Small [58]	13.7	1.9	79.6	-
ResT-Base [58]	30.3	4.3	81.6	-
ResT-Large [58]	51.6	7.9	83.6	-

Method	#Params. (M)	FLOPs (G)	Acc. 1K	Acc. 21K/Distill.
ViTC-1GF [62]	4.6	1.1	75.3	-
ViTC-4GF [62]	17.8	4.0	81.4	81.2†
ViTC-18GF [62]	81.6	17.7	83.0	84.9†
ViTC-36GF [62]	167.8	35	84.2	85.8†
CoAtNet-0 [40]	25	4.2	81.6	-
CoAtNet-1 [40]	42	8.4	83.3	-
CoAtNet-2 [40]	75	15.7	84.1	87.1†
CoAtNet-3 [40]	168	34.7	84.5	87.6†
CoAtNet-4↑384 [40]	275	189.5	-	88.4†
TNT-S [60]	23.8	5.2	81.3	-
TNT-B [60]	65.6	14.1	82.8	-
TNT-S↑384 [60]	23.8	-	83.1	-
TNT-B↑384 [60]	65.6	-	83.9	-
Swin-T [36]	29	4.5	81.3	-
Swin-S [36]	50	8.7	83.0	-
Swin-B [36]	88	15.4	83.3	85.2†
Swin-L↑384 [36]	197	104	-	86.4†
VOLO-D1 [45]	27	6.8	84.2	-
VOLO-D2 [45]	59	14.1	85.2	-
VOLO-D3 [45]	86	20.6	85.4	-
VOLO-D4 [45]	193	43.8	85.7	-
VOLO-D5 [45]	296	69.0	86.1	-
VOLO-D3↑448 [45]	86	67.9	86.3	-
VOLO-D4↑448 [45]	193	197	86.8	-
VOLO-D5↑448 [45]	296	304	87.0	-

Method	#Params. (M)	FLOPs (G)	Acc. 1K	Acc. 21K/Distill.
T2T-ViTt-14 [64]	21.5	6.1	81.7	-
T2T-ViTt-19 [64]	39.2	9.8	82.2	-
T2T-ViTt-19 [64]	64.1	15.0	82.6	-
PVT-Ti [42]	13.2	1.9	75.1	-
PVT-S [42]	24.5	3.8	79.8	-
PVT-M [42]	44.1	6.7	81.2	-
PVT-L [42]	61.4	9.8	81.7	-
PVTv2-B2 [66]	25.4	4.0	82.0	-
PVTv2-B4 [66]	62.6	10.1	83.6	-
CvT-13 [37]	20	4.5	81.6	-
CvT-21 [37]	32	7.1	82.5	-
CvT-13↑384 [37]	20	16.3	83.0	83.3†
CvT-21↑384 [37]	32	24.9	83.3	84.9†
CvT-W24↑384 [37]	277	193.2	-	87.7†
PiT-Ti [65]	4.9	0.7	73.0	74.6Ŧ
PiT-XS [65]	10.6	1.4	78.1	79.1Ŧ
PiT-S [65]	23.5	2.9	80.9	81.9Ŧ
PiT-B [65]	73.8	12.5	82.0	84.0Ŧ
CrossViT-9 [180]	8.6	1.8	73.9	-
CrossViT-15 [180]	27.4	5.8	81.5	-
CrossViT-18 [180]	43.3	9.0	82.5	-
CrossViT-15*↑384 [180]	28.5	21.4	83.5	-
CrossViT-18*↑384 [180]	44.6	32.4	83.9	-

Method	#Params. (M)	FLOPs (G)	Acc. 1K	Acc. 21K/Distill.
DeepViT-S [67]	27	6.2	82.3	-
DeepViT-L [67]	55	12.5	83.1	-
CaiT-XS-24 [43]	26.6	5.4	81.8	82.0Ŧ
CaiT-S-24 [43]	46.9	9.4	82.7	83.5Ŧ
CaiT-S-36 [43]	68.2	13.9	83.3	84.0Ŧ
CaiT-M-24 [43]	185.9	36.0	83.4	84.7Ŧ
CaiT-M-36 [43]	270.9	53.7	83.8	85.1Ŧ
CaiT-S-36↑384 [43]	68.2	48.0	84.0	85.4Ŧ
CaiT-M-36↑384 [43]	270.9	173.3	84.9	86.1Ŧ
DiversePatch-S12 [68]	22	-	81.2	-
DiversePatch-S24 [68]	44	-	82.2	-
DiversePatch-B12 [68]	86	-	82.9	-
DiversePatch-B24 [68]	172	-	83.3	-
DiversePath-B12↑384 [68]	86	-	84.2	-
Refined-ViT-S [38]	25	7.2	83.6	-
Refined-ViT-M [38]	55	13.5	84.6	-
Refined-ViT-L [38]	81	19.1	84.9	-
Refined-ViT-M↑384 [38]	55	49.2	85.6	-
Refined-ViT-L↑384 [38]	81	69.1	85.7	-
LV-ViT-S [44]	26	6.6	83.3	-
LV-ViT-M [44]	56	16.0	84.0	-
LV-ViT-L↑288 [44]	150	59.0	85.3	-
LV-ViT-M↑384 [44]	56	42.2	85.4	-
LV-ViT-L↑448 [44]	150	157.2	85.9	-

freedom for channel adjustment, and 2) last few class-attention stages with frozen patch embeddings. A later class token is inserted to model global representations, similar to DEtection with TRansformer (DETR) [30] with an encoder–decoder structure. This explicit separation is based on the assumption that the class token is invalid for the gradient of patch embeddings in the forward pass. With distillation training strategy [40], CaiT achieves a new SoTA on imagenet-1k (86.5% top-1 accuracy) without external data. Although deep Transformer suffers from attention collapse and oversmoothing problems, it still largely preserves the diversity of the attention map between different heads. Based on this observation, Zhou et al. [66] proposed Deep ViT that aggregates different head attention maps and regenerates a new one by using a linear layer to increase cross-layer feature diversity. Furthermore, Refiner [37] applies a linear layer to expand the dimension of the attention maps (indirectly increasing the head number) for diversity promotion. Then, a distributed local attention (DLA) is employed to achieve better modeling of both the local features and the global ones, which is implemented by a headwise convolution effecting on the attention map.

From the aspect of training strategy, Gong et al. [67] presented three patch diversity losses for deep Transformer that can significantly encourage patches' diversity and offset oversmoothing problem. Similar to [176], a patchwise cosine loss minimizes pairwise cosine similarity among patches. A patchwise contrastive loss regularizes the deeper patches by their corresponding one in the early layer. Inspired by Cutmix [177], a patchwise mixing loss mixes two different images and forces each patch to only attend to the patches from the same image and ignore unrelated ones. Distinct from the similar loss function of LV-ViT [43], it is motivated by patch diversity rather than patch-based label augmentation that LV-ViT [43] focuses on.

G. Transformers With Self-Supervised Learning

Following the grateful success of self-supervised learning (SSL) in the NLP field [5], recent works also attempt to design various self-supervised learning schemes for the visual Transformers in both generative and discriminative ways.

For the generative models, Chen et al. [68] proposed an image GPT (iGPT) for self-supervised visual learning. Different from the patch embedding of ViT [29], iGPT directly resizes and flattens the image to a lower resolution sequence. The resized sequences are then input into a GPT-2 [4] for autoregressive pixel prediction. iGPT demonstrates the effectiveness of the Transformer in the visual tasks without any help from image-specific knowledge, but its considerable computation cost is hard to be accepted (roughly 2500 V100-days for pretraining). Instead of the pixel-wise generation Bao et al. [5] proposed a BERT-style visual Transformer (BEiT) [70] by reconstructing the masked image in the latent space. Precisely, a dVAE [148] first converts input patches into discrete visual tokens, like BERT's [5] dictionary. These tokens are then employed as latent pseudo labels for SSL.

For the discriminative models, Chen et al. [72] investigated the effects of several fundamental components for stabilized self-supervised ViT training. They observed that the unstable training process mildly affects the eventual performance and extended MoCo series to MoCo v3, containing a series of training strategies such as freezing projection layer. Following DeiT [40], Caron et al. [73] further extended the teacher–student recipe to self-supervised learning and propose DINO. The core concepts of DINO can be summarized into three points. A momentum encoder inherited SwAV [178] serves as a teacher model that outputs the centered pseudo labels over a batch. An online encoder without the prediction head serves as a student model to fit the teacher's output. A standard cross-entropy loss connects self-training with knowledge distillation. More interestingly, self-supervised ViT can learn flourishing features for segmentation, which are normally unattainable by the supervised models.

H. Discussion

1) Algorithm Evaluation and Comparative Analysis: In our taxonomy, all the existing supervised models are grouped into six categories. Table I summarizes the performances of these existing visual Transformers on ImageNet-1k benchmarks. To evaluate them objectively and intuitively, we use the following three figures to illustrate their performances on ImageNet-1k under different configurations. Fig. 8(a) summarizes the accuracy of each model under 224^2 inputs size. Fig. 8(b) takes the FLOPs as the horizontal axis, which focuses on their performances under higher resolution. Fig. 8(c) focuses on the pretrained models with external datasets. From these

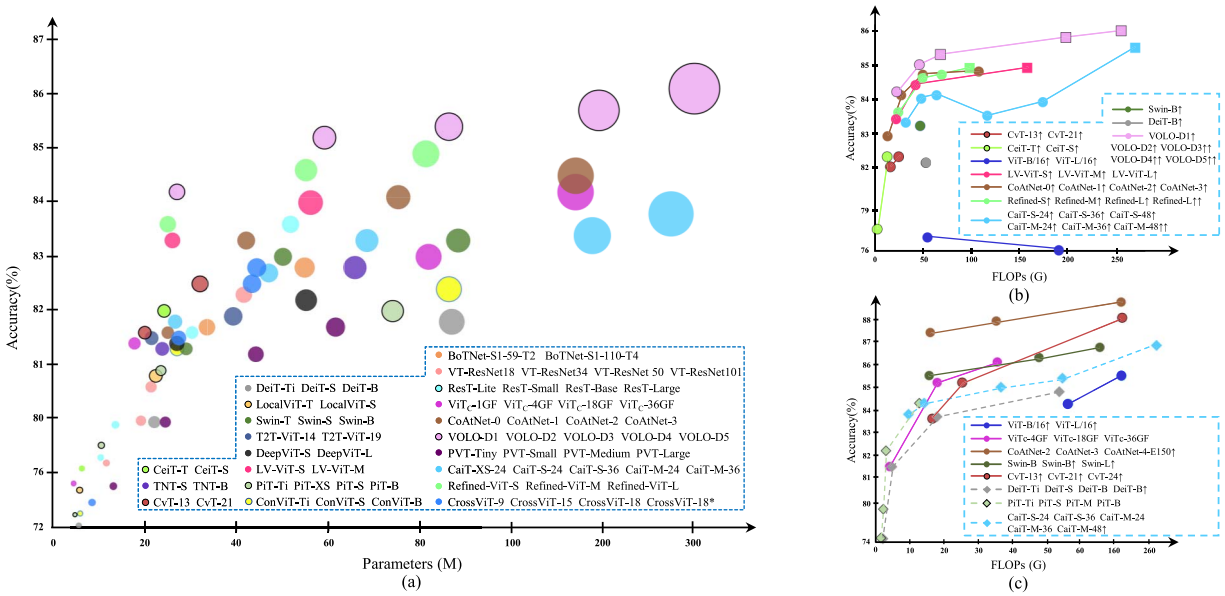

Fig. 8. Comparisons of recent visual Transformers on ImageNet-1k benchmark, including [29], [36], [37], [41], [42], [43], [50], [51], [52], [53], [54], [58], [62], [63], [65], [180] (best viewed in color). (a) Bubble plot of the mentioned models with 224^2 resolution input, the size of cycle denotes GFLOPs. (b) Comparison on high-resolution inputs, the square indicates 448^2 input resolution. (c) Accuracy plot of some pretrained models on ImageNet-21k.

comparison results, we briefly summarize several performance improvements on efficiency and scalability as follows.

1) Compared with the most structure-improved methods, the basic training strategies, such as DeiT [40] and LV-ViT [43], are more universal for various models.
2) The locality is indispensable for the Transformer, which is reflected by the dominant of VOLO [44] and Swin [35] on various tasks.
3) The convolutional patchify stem (ViT_c [58]) and early convolutional stage (CoAtNet [39]) can significantly boost the accuracy of the Transformers, especially for large models. We speculate that the reason is because they introduce a more stringent high-level feature than the sketchy patch projection in ViT [29].
4) The deep Transformer, such as Refined-ViT [37] and CaiT [42], has great potential. As the model size grows quadratically with the channel dimension, the tradeoff in deep Transformer is considered for further investigation.
5) CeiT [54] and CvT [36] show significant advantages in training a small or medium model (0–40M), which suggests that such kinds of hybrid attention blocks for lightweight models are worth further exploring.

2) Brief Discussion on Alternatives: During the development of the visual Transformers, the most common question is whether the visual Transformers can replace the traditional convolution completely. By reviewing the history of the performance improvements in the last year, there is no sign of relative inferiority here. The visual Transformers have returned from a pure structure to a hybrid form, and the global information has gradually returned to a mixed stage with the locality bias. Although the visual Transformers can be equivalent to CNN or even has a better modeling capability, such a simple and effective convolution operation is enough to process the locality and the semantic features in the shallow layer. In the future, the spirit of combining both of them shall drive more breakthroughs for image classification.

IV. Transformer for Detection

In this section, we review visual Transformers for object detection, which can be grouped into two folds: Transformer as the neck and Transformer as the backbone. For the neck detectors, we mainly focus on a new representation specified to the Transformer structure, called object query, that a set of learned parameters aggregate instance features from input images. The recent variants try to solve an optimal fusion paradigm in terms of either convergence acceleration or performance improvement. Besides these neck designs, a proportion of backbone detectors also take specific strategies into consideration. Finally, we evaluate them and analyze some potential methods for these detectors.

A. Transformer Neck

We first review DETR [30] and Pix2seq [75], two original Transformer detectors based on different paradigms. Subsequently, we mainly focus on the DETR-based variants, improving Transformer detectors in accuracy and convergence from five aspects: sparse attention, spatial prior, structural redesign, assignment optimization, and pretraining model.

1) Original Detectors: DETR [30] is the first end-to-end Transformer detector that eliminates hand-designed representations [180], [181], [182], [183] and nonmaximum suppression (NMS) postprocessing, which redefines the object detection as a set prediction problem. As shown in Fig. 9, a small set of learnable positional encodings, called object queries, are parallelly fed into the Transformer decoder to extract the instance information from the image features. Then, these object queries are independently predicted to be a detection result. Instead of the vanilla k-class classification, a special class, no object label ($\varnothing$) is added for k + 1 class classification. During the training process, a bipartite matching strategy is applied between the predicted objects and the ground truth to identify one-to-one label assignment, hence removing the redundant predictions at the inference time without NMS. In backpropagation, a Hungarian loss includes

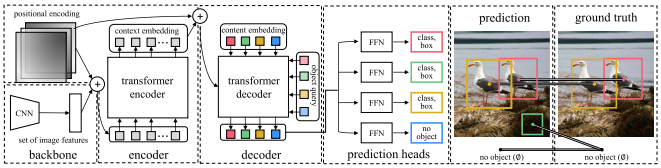

Fig. 9. Overview of DETR. (Modified from [30].)

a log-likelihood loss for all classification results and a box loss for all the matched pairs. More details about the Hungarian matching strategy are available in the Supplementary Material.

Overall, DETR provides a new paradigm for end-to-end object detection. The object query gradually learns an instance representation during the interaction with image features. The bipartite matching allows a direct set prediction and easily joints to the one-to-one label assignment, hence eliminating traditional postprocessing. DETR achieves competitive performance on the COCO benchmark but suffers from slow convergence as well as poor performance on small objects.

Another pioneered work is Pix2seq [75], treating generic object detection as a language modeling task. Given an image input, a vanilla sequential Transformer is executed to extract features and generate a series of object descriptions (i.e., class labels and bounding boxes) autoregressively. Such a simplified but more elaborate image caption method is derived under the assumption that if a model learns about both location and label of an object, it can be taught to produce a description with specified sequence [75]. Compared with DETR, Pix2seq attains a better result on small objects. How to combine both kinds of concepts is worthy of further consideration.

2) Transformer With Sparse Attention: In DETR, the dense interaction across both object queries and image features costs unbearable resources and slows down the convergence of DETR. Therefore, the most recent efforts aim to design data-dependent sparse attention to address these issues.

Following [184], Zhu et al. [76] developed deformable DETR to ameliorate both training convergence and detection performance significantly via multiscale deformable attention. Compared with the original DETR, the deformable attention module only samples a small set of key (reference) points for full feature aggregation. Such sparse attention can be easily stretched to multiscale feature fusion without the help of FPN [185]. Moreover, an iterative bounding box refinement and a two-stage prediction strategy (Section IV-A3) are developed to further enhance the detection accuracy. Empirically, deformable DETR achieves a higher accuracy (especially for small objects) with $10\times$ less training epochs and reduces the computing complexity to $O(2N_qC^2+\min(\mathrm{HWC}^2, N_{\mathrm{qKC}}^2))$ with $1.6\times$ faster inference speed. Please see the Supplementary Material for more details of deformable attention mechanism.

By visualizing the encoder attention maps of DETR [30], Zheng et al. [78] observed that the semantically similar and spatially close elements always have similar attention maps. As such, they presented an adaptive clustering Transformer (ACT), leveraging a multiround sensitivity hashing to dynamically cluster object queries into different prototypes. The attention map of prototypes is then broadcast to their corresponding queries. Unlike the redesign on sparse attention, Wang et al. [78] introduced a poll and pool (PnP) sampling model to extract the fine foreground features and condense the contextual background features into a smaller one. Such fine–coarse tokens are then fed into DETR to generate the detection results. Instead of the input sparsification, Sparse DETR [79] applied a hysteretic scoring network (corresponding to the poll operation in [78]) to update the expected tokens selectively within the transformer encoder, where the top-k selected tokens are supervised by pseudo labels from the binarized decoder cross-attention map with BCE loss.

3) Transformer With Spatial Prior: Unlike anchor or other representations directly generated by content and geometry features [180], [186], object queries implicitly model the spatial information with random initialization, which is weakly related to the bounding box. The mainstream for spatial prior applications is the one-stage detector with empirical spatial information and the two-stage detector with geometric coordinates initialization or region-of-interest (RoI) features.

In one-stage methods, Gao et al. [80] suggested spatially modulated cross attention (SMCA) to estimate the object queries' spatial prior explicitly. Specifically, a Gaussian-like weight map generated by object queries is multiplied with the corresponding cross-attention map to augment the RoI for convergence acceleration. Furthermore, both intrascale and multiscale self-attention layers are utilized in the Transformer encoder for multiscale feature aggregation, and the scale-selection weights generated from object queries are applied for scale-query adaptation. Meng et al. [81] extracted the spatial attention map from the cross-attention formulation and observe that the extreme region of such attention map has larger deviations at the early training. Consequently, they proposed conditional DETR where a new spatial prior mapped from reference points is adopted in the cross-attention layer, thereby attending to extreme regions of each object explicitly. The reference point is predicted by the object query or serves as a learned parameter replacing the object query. Following [81], Anchor DETR [82] suggests to explicitly learn the 2-D anchor points ($[cx, cy]$) and different anchor patterns instead of the high-dimensional spatial embedding. Similar to [180], the pattern embeddings are assigned to the meshed anchor points so that they can detect different scale objects anywhere. DAB-DETR [83] then extends the 2-D concept to a 4-D anchor box ($[cx, cy, w, h]$) to explicitly provide proposal bounding box information during the cross attention. With the auxiliary decoder loss of the coordinates offset [76], such a 4-D box can be dynamically refined layer-by-layer in the decoder. However, the same reference boxes/points may severely deteriorate queries' saliency and confuses the detector due to the indiscriminative spatial prior. By assigning query-specific reference points to object queries, SAP-DETR [84] only predicts the distance from each side of the bounding box to these points. Such a query-specific prior discrepancies queries' saliency paves the way for fast model convergence.

In two-stage methods, Zhe et al. [76] empowered the Top-K region proposals from encoder features to initialize the decoder embedding instead of the learned parameters. Efficient DETR [85] also adopts a similar initialization operation for dense proposals and refines them in the decoder to get sparse prediction by using a shared detection head with the dense parts. More interestingly, it is observed that small stacking decoder layers bring slight performance improvement, but

more stacks yield even worse results. Dynamic DETR [86] regards the object prediction in a coarse-to-fine process. Different from the previous RoI-based initialization, according to queries' reference boxes, a query-based weight is used to replace cross-attention layers and directly affect their corresponding coarse RoI features for query refinement.

4) Transformer With Redesigned Structure: Besides the modifications focusing on the cross attention, some works redesign an encoder-only structure to avoid the problem of the decoder directly. TSP [87] inherits the idea of set prediction [30] and dismisses the decoder and the object query to accelerate convergence. Such encoder-only DETR reuses previous representations [180], [186] and generates a set of fixed-size features of interests (FoIs) [186] or proposals [180] that are subsequently fed into the Transformer encoder. In addition, a matching distillation is applied to resolve the early instability of the bipartite matching during training process. Fang et al. [88] presented an encoder-only decoder YOLOS, a pure sequence-to-sequence Transformer to unify the classification and detection tasks. It inherits ViT's structure and replaces the single class token with fixed-size learned detection tokens. These object tokens are first pretrained on the transfer ability for the classification tasks and then fine-tuned on the detection benchmark.

5) Transformer With Bipartite Matched Optimization: In DETR [30], the bipartite matching strategy forces the prediction results to fulfill one-to-one label assignment during the training scheme. Such a training strategy simplifies detection pipeline and directly builds up an end-to-end system without the help of NMS. To deeply understand the efficacy of the end-to-end detector, Sun et al. [189] devoted to exploring a theoretical view of one-to-one prediction. Based on multiple ablation and theoretical analyses, they concluded that the classification cost for one-to-one matching strategy serves as the key component for significantly avoiding duplicate predictions. Even so, DETR is suffering from multiple problems caused by bipartite matching. Li et al. [91] exploited a denoising DETR (DN-DETR) to mitigate the instability of bipartite matching. Concretely, a series of objects with slight perturbation is supposed to denoise to their original coordinates. The main ingredients of the denoising part are an attention mask that prevents information leakage between the matching and noised parts, and a specified label embedding to indicate the perturbation. Recently, Zhang et al. [92] presented an improved denoising training model called DINO (2022) by incorporating a contrastive loss for the perturbation groups. Based on DN-DETR [91], DINO attaches a "no object" class for the negative example if the distance is far enough from the perturbation, which avoids redundant prediction due to the confusion of multiple reference points near an object. As a result, DINO attains the current SoTA on the COCO dataset.

6) Transformer Detector With Pretraining: Inspired by the pretrained linguistic Transformer [3], [5], Dai et al. [89] devised an unsupervised pretraining DETR (UP-DETR) to assist the convergence for supervised training. The objective of pretraining is to localize the random cropped patches from a given image. Specifically, each patch is assigned to a set of queries and predicted independently via the attention mask. An auxiliary reconstruction loss forces the detector to preserve the feature discrimination so as to avoid overbias toward the localization in pretraining. FP-DETR [90] devotes to narrowing the gap between upstream and downstream tasks. During the pretraining, a fully encoder-only DETR such as YOLOS [88] views query positional embeddings as a visual prompt to enhance target area attention and object discrimination. A task adapter implemented by self-attention is used to enhance object interaction during fine-tuning.

B. Transformer Backbone

We have reviewed numerous Transformer-based backbones for image classification [29], [40] in Section III. These backbones can be easily incorporated into various frameworks [30], [182], [187] to perform dense prediction tasks. For example, the hierarchical structure, such as PVT [41], [65], constructs the visual Transformer as a high-to-low resolution process to learn multiscale features. The locally enhanced structure constructs the backbone as a local-to-global combination, which can efficiently extract both short- and long-range visual dependencies and avoid quadratic computational overhead, such as Swin-Transformer [35], ViL [61], and Focal Transformer [62]. The Supplementary Material includes more detailed comparisons of these models for the dense prediction tasks. In addition to the generic Transformer backbone, the feature pyramid Transformer (FPT) [93] combines the characteristics across both the spaces and the scales, by using self-attention, top-down cross attention, and bottom-up cross-channel attention. Following [190], HRFormer [94] introduces the advantages of multiresolution to the Transformer along with nonoverlapping local self-attention. HRViT [95] redesigns a heterogeneous branch and a cross-shaped attention block to further optimize the tradeoff between efficiency and accuracy.

C. Discussion

We summarize fivefold of the Transformer neck detectors in Table II, and more details of Transformer backbone for dense prediction tasks are referred to in Table SI of the Supplementary Material. The majority of Transformer neck promotions concentrate on the following five aspects.

1) The sparse attention model and the scoring network are proposed to address the problem of redundant feature interaction. These methods can significantly alleviate computational costs and accelerate model convergence.
2) The explicit spatial prior, which is decomposed into the selected feature initialization and the positional information extracted by learned parameters, would enable the detector to predict the results precisely.
3) Multiscale features and iterative box refinement are benefit DETR for small object detection.
4) The improved bipartite matching strategy is beneficial to avoid redundant prediction, add positive gradients, and perform end-to-end object detection.
5) The encoder-only structure reduces the overall Transformer stack layers but increases the FLOPs excessively, while the encoder–decoder structure is a good tradeoff between FLOPs and parameters, but the deeper decoder layers may cause the slow convergence.

Existing Transformer backbones mostly focus on the classification task, but a few works are developed for the

TABLE II

COMPARISON BETWEEN TRANSFORMER NECKS AND REPRESENTATIVE CNNS WITH RESNET-50 BACKBONE ON COCO 2017 VAL SET

Method	Epochs	FLOPs (G)	#Para. (M)	FPS	MS	AP/AP$_{50}$/AP$_{75}$	Ap$_S$/Ap$_M$/Ap$_L$
CNN Backbone with Other Representations							
FCOS [88], [187]	36	177	-	17	✓	41.0 /59.8/44.1	26.2/44.6/52.2
Faster R-CNN [181]	37	180	42	26	✓	40.2/61.0/43.8	24.2/43.5/52.0
Faster R-CNN+ [181]	109	180	42	26	✓	42.0 /62.1/45.5	26.6/45.4/53.4
Mask R-CNN [188]	36	260	44	-	✓	41.0/61.7/44.9	- / - / -
Cas. Mask R-CNN [189]	36	739	82	18	✓	46.3/64.3/50.5	- / - / -
Transformer Model as Neck							
DETR-R50 [30]	500	86	41	28	✗	42.0/62.4/44.2	20.5/45.8/61.1
DETR-DC5 [30]	500	187	41	12	✗	43.3/63.1/45.9	22.5/47.3/61.1
Pix2seq [76]	300	-	37	-	✗	43.0/61.0/45.6	25.1/46.9/59.4
Pix2seq-DC5 [76]	300	-	38	-	✗	43.2/61.0/46.1	26.6/47.0/58.6
Defor. DETR [77]	50	78	34	23	✗	39.7/60.1/42.4	21.2/44.3/56.0
Defor. DETR-DC5 [77]	50	128	34	22	✗	41.5/61.8/44.9	24.1/45.3/56.0
Defor. DETR-Iter [77]	50	173	40	19	✓	43.8/62.6/47.7	26.4/47.1/58.0
Defor. DETR-Two [77]	50	173	40	19	✓	46.2/65.2/50.0	28.8/49.2/61.7
ACT-DC5 (L=16) [78]	MTKD [78]	156	-	14	✗	40.6/ - / -	18.5/44.3/59.7
ACT-DC5 (L=32) [78]	MTKD [78]	169	-	16	✗	43.1/ - / -	22.2/47.1/61.4
PnP-DETR-0.33 [79]	500	77	-	-	✗	41.1/61.5/43.7	20.8/44.6/60.0
PnP-DETR-0.5 [79]	500	79	-	-	✗	41.8/62.1/44.4	21.2/45.3/60.8
PnP-DETR-DC5-0.5 [79]	500	136	-	-	✗	43.1/63.4/45.3	22.7/46.5/61.1
Sparse-DETR-0.1 [80]	50	105	41	25	✓	45.3/65.8/49.3	28.4/48.3/60.1
Sparse-DETR-0.5 [80]	50	136	41	21	✓	46.3/66.0/50.1	29.0/49.5/60.8
SMCA [81]	50	86	40	22	✗	41.0/ - / -	21.9/44.3/59.1
SMCA+ [81]	108	86	40	22	✗	42.7/ - / -	22.8/46.1/60.0
SMCA [81]	50	152	40	10	✓	43.7/63.6/47.2	24.2/47.0 /60.4
SMCA+ [81]	108	152	40	10	✓	45.6/65.5/49.1	25.9/49.3/62.6
Condit. DETR [82]	108	90	44	17	✗	43.0/64.0/45.7	22.7/46.7/61.5
Condit. DETR-DC5 [82]	108	195	44	11	✗	45.1/65.4/48.5	25.3/49.0/62.2
Anchor-DETR [83]	50	85	39	20	✗	42.1/63.1/44.9	22.3/46.2/60.0
Anchor-DETR-DC5 [83]	50	151	39	14	✗	44.2/64.7/47.5	24.7/48.2/60.6
DAB-DETR [84]	50	90	44	17	✗	42.2/63.1/44.7	21.5/45.7/60.3
DAB-DETR-DC5 [84]	50	194	44	11	✗	44.5/65.1/47.7	25.3/48.2/62.3
SAP-DETR [85]	50	92	47	16	✗	43.1/63.8/45.4	22.9/47.1/62.1
SAP-DETR-DC5 [85]	50	197	47	9	✗	46.0/65.5/48.9	26.4/50.2/62.6
Efficient DETR [86]	36	159	32	-	✓	44.2/62.2/48.0	28.4/47.5/56.6
Efficient DETR* [86]	36	210	35	-	✓	45.1/63.1/49.1	28.3/48.4/59.0
Dynamic DETR [87]	50	-	58	-	✓	47.2/65.9/51.1	28.6/49.3/59.1
TSP-FCOS [88]	36	189	52	15	✓	43.1/62.3/47.0	26.6/46.8/55.9
TSP-RCNN [88]	36	188	64	11	✓	43.8/63.3/48.3	28.6/46.9/55.7
TSP-RCNN+ [88]	96	188	64	11	✓	45.0/64.5/49.6	29.7/47.7/58.0
YOLOS-S(800×) [89]	150	194	31	6	✗	36.1/56.4/37.1	15.3/38.5/56.1
YOLOS-S(784×) [89]	150	172	28	6	✓	37.6/57.6/39.2	15.9/40.2/57.3
YOLOS-B [89]	150	538	127	3	✗	42.0/62.2/44.5	19.5/45.3/62.1
UP-DETR [90]	150	86	41	28	✗	40.5/60.8/42.6	19.0/44.4/60.0
UP-DETR+ [90]	300	86	41	28	✗	42.8/63.0/45.3	20.8/47.1/61.7
FP-DETR-Base [91]	50	-	36	-	✗	43.3/63.9/47.7	27.5/46.1/57.0
DN-DETR [92]	50	94	44	17	✗	44.1/64.4/46.7	22.9/48.0/63.4
DN-DETR-DC5 [92]	50	202	44	8	✗	46.3/66.4/49.7	26.7/50.0/64.3
DN-Defor.-DETR [92]	50	196	48	23	✓	46.3/66.4/49.7	26.7/50.0/64.3
DINO-4scale [93]	36	279	47	24	✓	50.5/68.3/55.1	32.7/53.9/64.9
DINO-5scale [93]	36	860	47	10	✓	51.0/69.0/55.6	34.1/53.6/65.6

"MS" denotes to multi-scale features.
Both GFLOPs and Params are measured by Detectron2. FPS is measured on a single A100 GPU.

dense prediction tasks. In the future, we anticipate that the Transformer backbone would cooperate with the deep high-resolution network to solve dense prediction tasks.

V. TRANSFORMER FOR SEGMENTATION

Patch- and query-based Transformers are the two major ways for segmentation. The latter can be further grouped into object query and mask embedding methods.

A. Patch-Based Transformer

Because of the receptive field expansion strategy [191], CNNs require multiple decoder stacks to map the high-level features into the original spatial resolution. Instead, patch-based Transformer can easily incorporate with a simple decoder for segmented mask prediction because of its global modeling capability and resolution invariance. Zheng et al. extended ViT [29] for semantic segmentation tasks and presented SEgmentation TRansformer (SETR) [96] by employing three fashions of the decoder to perform per-pixel classification: naive upsampling, progressive upsampling, and multilevel feature aggregation (MLA). SETR demonstrates the feasibility of the visual Transformer for the segmentation tasks, but it also brings unacceptably extra GPU costs. TransUNet [97] is the first for medical image segmentation. Formally, it can be viewed as either a variant of SETR with MLA decoder [96] or a hybrid model of U-Net [192] and Transformer. Due to the strong global modeling capability of Transformer encoder, Segformer [98] designs a lightweight decoder with only four MLP layers. Segformer shows superior performance as well as stronger robustness than CNNs when tested with multiple corrupted types of images.

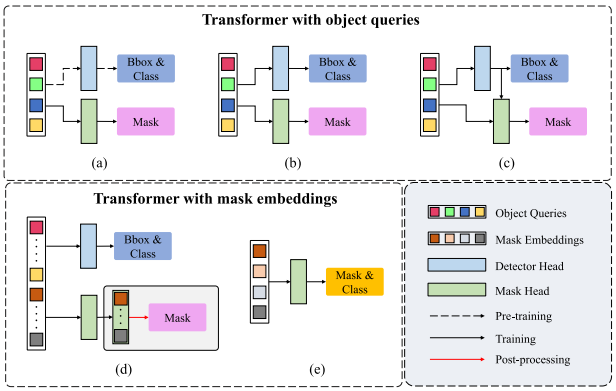

Fig. 10. Query-based frameworks for segmentation tasks. (a) Transfer learning for fine-tuning mask head. (b) Multitask learning for two independent tasks. (c) Cascade learning predict fine-grained masks based on box results. (d) Query embeddings are independently supervised by mask embeddings and boxes. (e) Box-free model directly predicts masks without box branch and views segmentation task as a mask prediction problem.

B. Query-Based Transformer

Query embeddings are a set of scratch semantic/instance representations gradually learning from the image inputs. Unlike patch embeddings, queries can more "fairly" integrate the information from features and naturally join with the set prediction loss [30] for postprocessing elimination. The existing query-based models can be grouped into two categories. One (object query) is driven by both detection and segmentation tasks, simultaneously. The other (mask embedding) is only supervened by segmentation task.

1) Object Queries: There are three training manners for object query-based methods [Fig. 10(a)–(c)]. With the success of DETR [30] for the object detection tasks, the authors extend it to panoptic segmentation (hence termed panoptic DETR [30]) by training a mask head based on the pre-trained object queries [Fig. 10(a)]. In detail, a cross-attention block between the object queries and the encoded features is applied to generate an attention map for each object. After an upsampling FPN-style CNN, a spatial argmax operation fuses the resulting binary masks to a nonoverlapping prediction. Instead of using a multistage serial training process, Cell-DETR and VisTR develop a parallel model for end-to-end instance segmentation [Fig. 10(b)]. Based on DETR [30], Cell-DETR leverages a cross-attention block to extract instancewise features from the box branch and fuses the previous backbone features to augment the CNN decoder for accurate instance mask segmentation of biological cells. Another extension is VisTR [100] that directly formulates the video instance segmentation (VIS) task as parallel sequence prediction. Apart from the similar structure as Cell-DETR [99], the key of VisTR is a bipartite matching loss at the instance sequence level to maintain the order of outputs, so as to adapt DETR [30] to VIS for direct one-to-one predictions. Unlike prior works

TABLE III

COMPARISON BETWEEN CNN- AND TRANSFORMER-BASED MODEL ON ADE20K AND COCO FOR DIFFERENT SEGMENTATION TASKS. "+MS" DENOTES THE MULTISCALE INPUTES

(a) ADE20K Val. Set for Semantic Segmentation

Method	Backbone	image size	#Params. (M)	FLOPs (G)	FPS	mIoU	+MS
UperNet [32] [195] [196]	R-50 [11]	512	67	238	23.4	42.1	42.8
	R-101 [11]	512	86	257	20.3	43.8	44.9
	Swin-T [36]	512	60	236	18.5	44.5	46.1
	Swin-S [36]	512	81	259	15.2	47.6	49.3
	Swin-B† [36]	640	121	471	8.7	50.0	51.6
	Swin-L† [36]	640	234	647	6.2	52.0	53.5
Segformer [99]	MiT-B3	512	47	79	-	49.4	50.0
	MiT-B4	512	64	96	15.4	50.3	51.1
	MiT-B5	640	85	183	9.8	51.0	51.8
Segmenter [105]	ViT-S/16† [29]	512	27	-	34.8	45.3	46.9
	ViT-B/16† [29]	512	106	-	24.1	48.5	50.0
	ViT-L/16† [29]	640	334	-	-	51.8	53.6
MaskFormer [32]	R-50 [11]	512	41	53	24.5	44.5	46.7
	R-101 [11]	512	60	73	19.5	45.5	47.2
	Swin-T [36]	512	42	55	22.1	46.7	48.8
	Swin-S [36]	512	63	79	19.6	49.8	51.0
	Swin-B† [36]	640	102	195	12.6	52.7	53.9
	Swin-L† [36]	640	212	375	7.9	54.1	55.6

(b): COCO Test-Dev for Instance Segmentation

Method	Backbone	Epochs	AP^{box}/AP^{seg}	AP^{seg}_S	AP^{seg}_M	AP^{seg}_L	FPS
Mask R-CNN [188]	R-50-FPN [11]	36	41.3/37.5	21.1	39.6	48.3	15.3
	R-101-FPN [11]	36	43.1/38.8	21.8	41.4	50.5	11.8
Blend Mask [197]	R-50-FPN [11]	36	43.0/37.8	18.8	40.9	53.6	15.0
	R-101-FPN	36	44.7/39.6	22.4	42.2	51.4	11.5
SOLO v2 [198]	R-50-FPN [11]	36	40.7/38.2	16.0	41.2	55.4	10.5
	R-101-FPN [11]	36	42.6/39.7	17.3	42.9	57.4	9.0
ISTR [103]	R-50-FPN [11]	36	46.8/38.6	22.1	40.4	50.6	13.8
	R-101-FPN [11]	36	48.1/39.9	22.8	41.9	52.3	11.0
SOLQ [104]	R-50 [11]	50	47.8/39.7	21.5	42.5	53.1	-
	R-101 [11]	50	48.7/40.9	22.5	43.8	54.6	-
	Swin-L† [36]	50	55.4/45.9	27.8	49.3	60.5	-
QueryInst [102]	R-50-FPN [11]	36	44.8/40.1	23.3	42.1	52.0	10.5
	R-50-FPN [11]	36	45.6/40.6	23.4	42.5	52.8	7.0
	R-101-FPN [11]	36	47.0/41.7	24.2	43.9	53.9	6.1
	Swin-L† [36]	50	56.1/49.1	31.5	51.8	63.2	3.3

(c): COCO Panopticon Minival. for Panoptic Segmentation

Method	Backbone	Epochs	#Params. (M)	FLOPs (G)	PQ	PQ^{Th}	PQ^{St}
DETR [30]	R-50 [11]	500+25	43	137	43.4	48.2	36.3
	R-101 [11]		62	157	45.1	50.5	37.0
MaxDeepLab [31]	Max-S	54	62	162	48.4	53.0	41.5
	Max-L		451	1846	57.0	42.2	51.1
MaskFormer [199]	R-50 [11]	300	45	181	46.5	51.0	39.8
	R-101 [11]		64	248	47.6	52.5	40.3
	Swin-T [36]		42	179	47.7	51.7	41.7
	Swin-S [36]		63	259	49.7	54.4	42.6
	Swin-B [36]		102	411	51.1	56.3	43.2
	Swin-L† [36]		212	792	52.7	58.5	44.0

† denotes the model pre-trained on ImageNet-21k

that treat detection and mask generation branches separately, QueryInst [101] builds a hybrid cascaded network [Fig. 10(c)], where the previous box outputs together with the shared queries serve as the inputs of the mask head for accurate mask segmentation. Notably, QueryInst leverages the shared queries to keep the instance correspondences across multistage so that mitigating the problem of inconsistent objects in previous nonquery-based methods [188], [193]. QueryInst obtains the latest SoTA results on the COCO datasets.

2) Mask Embeddings: The other framework makes efforts to use queries to predict mask directly, and we refer to this learned mask-based query as mask embeddings. Unlike object queries, mask embeddings are only supervised by the segmentation tasks. As shown in Fig. 10(d), two disjoint sets of queries are employed parallelly for different tasks, and the box learning is viewed as an auxiliary loss for further enhancement. For semantic and box-free instance segmentation, a series of query-based Transformers predict the mask directly without the help of the box branch [Fig. 10(e)].

From the auxiliary training perspective, the core is how to enable 1-D sequence outputs to be supervised by 2-D mask labels directly. To this end, ISTR [102] empowered a mask precoding method to encode the ground-truth mask into a sparse mask vector for instance segmentation. Similarly, Dong et al. [103] proposed a more straightforward pipeline SOLQ

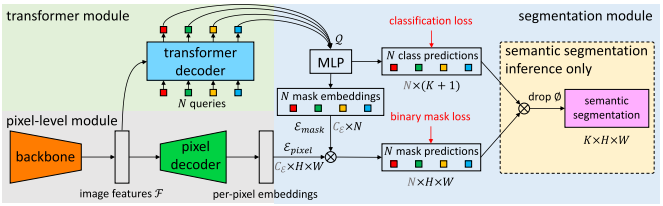

Fig. 11. Illustration of Maskformer (from [32]).

and explored three reversible compression encoding methods. In detail, a set of unified queries is applied to perform multiple representation learning parallelly: classification, box regression, and mask encoding. Based on the original DETR [191], SOLQ adds a mask branch to produce mask embedding loss. Both ISTR and SOLQ obtain comparable results and outperform previous methods even with approximation-based suboptimal embeddings. However, there exists a huge gap between AP^{box} and AP^{seg} (Table III).

From the box-free perspective, Wang et al. [31] pioneered a new paradigm Max-DeepLab that directly predicts panoptic masks from the query without the help of the box branch. Specifically, it forces the query to predict the corresponding mask via a PQ-style bipartite matching loss and a dual-path Transformer structure. Given a set of mask embeddings and an image input, Max-DeepLab processes them separately in both Transformer and CNN path and then generates a binary mask and a class for each query, respectively. Max-DeepLab achieves a new SoTA with 51.3% PQ on the COCO test-dev set but leads to heavy computational costs due to its dual-path high-resolution processing. Segmenter [104] views the semantic segmentation task as a sequence-to-sequence problem. In detail, a set of mask embeddings that represent different semantic classes are fed into the Transformer encoder together with image patches, and then, a set of labeled masks are predicted for each patch via an argmax operation.

Unlike the conventional pixel-wise segmentation, Cheng et al. [32] reformulated the semantic segmentation task as a mask prediction problem and enabled this output format to the query-based Transformer, which is called Maskformer. Different from Max-DeepLab [31], as shown in Fig. 11, Maskformer leverages a simple Transformer decoder without redundant connection as well as a sigmoid activation for overlapping binary mask selection. It not only outperforms the current per-pixel classification SoTA on large-class semantic segmentation datasets but also generalizes the panoptic segmentation task with a new SoTA result (Table III).

C. Discussion

We summarize the aforementioned Transformers according to three different tasks. Table III(a) focuses on ADE20K (170 classes). It can be shown that when trained on datasets with large numbers of classes, the segmentation performance of visual Transformers is improved significantly. Table III(b) focuses on the COCO test dataset for instance segmentation. Clearly, the visual Transformers with mask embeddings surpass most prevailing models for both segmentation and detection tasks. However, there is a huge performance gap between AP^{box} and AP^{seg}. With the cascaded framework,

QueryInst [101] attains the SoTA among various Transformer models. It is worthy of further study for combining the visual Transformers with the hybrid task cascade structures. Table III(c) focuses on panoptic segmentation. Max-DeepLab [31] is general to solve both foreground and background in the panoptic segmentation task via a mask prediction format. Maskformer [32] successfully employs this format and unifies both semantic and instance segmentation tasks into a single model. It is concluded that the visual Transformers could unify multiple segmentation tasks into one box-free framework with mask prediction.

VI. Transformer for 3-D Visual Recognition

With the rapid development of 3-D acquisition technology, stereo/monocular images and light detection and ranging (LiDAR) point clouds become the popular sensory data for 3-D recognition. Discriminated from the RGB(D) data, point cloud representation pays more attention to distance, geometry, and shape information. Notably, such a geometric feature is significantly suitable for Transformer on account of its characteristic on sparseness, disorder, and irregularity. Following the success of 2-D visual Transformers, substantial approaches are developed for 3-D visual analysis. This section exhibits a compact review for 3-D visual Transformers following representation learning, cognition mapping, and specific processing.

A. Representation Learning

Compared with conventional hand-designed networks, visual Transformer is more appropriate for learning semantic representations from point clouds, in which such irregular and permutation-invariant nature can be transformed into a series of parallel embeddings with positional information. Point Transformer [105] and PCT [106] first demonstrate the efficacy of the visual Transformer in 3-D scenes. The former merges a hierarchical Transformer [105] with the downsampling strategy [199] and extends their previous vector attention block [25] to 3-D point clouds. The latter first aggregates neighbor points and then processes such neighbor embeddings on a global offset Transformer where a knowledge transfer from graph convolution network (GCN) is applied for noise mitigation. Notably, the positional encoding, a significant operation of the visual Transformer, is diminished in both approaches because of points' inherent coordinate information. PCT directly processes on the coordinates without positional encodings, while Point Transformer adds a learnable relative positional encoding for further enhancement. Lu et al. [107] leveraged a local–global aggregation module 3DCTN to achieve local enhancement and cost efficiency. Given the multistride downsampling groups, an explicit graph convolution with max-pooling operation is used to aggregate the local information within each group. The resulting group embeddings are concatenated and fed into the improved transformer [105], [106] for global aggregation. Park et al. [108] presented Fast Point Transformer to optimize the model efficiency by using voxel-hashing neighbor search, voxel-bridged relative positional encoding, and similarity-based local attention.

For dense prediction, Pan et al. [109] proposed a customized point-based backbone Pointformer for attending the local and global interactions separately within each layer. Different from previous local–global forms, a coordinate refinement operation after the local attention is adopted to update the centroid point instead of the surface one. Also, a local–global cross-attention model fuses the high-resolution features, followed by global attention. Fan et al. [110] returned to the single-stride sparse transformer (SST) to address the problem for small-scale detection. Similar to Swin [35], a shifted group in continuous Transformer block is adopted to attend to each group of tokens separately, which further mitigates the computation problem. In voxel-based methods, voxel transformer (VoTr) [111] separately operate on the empty and nonempty voxel positions effectively via local attention and dilated attention blocks. VoxSeT [112] further decomposes the self-attention layer into two cross-attention layers, and a group of latent codes link them to preserve global features in a hidden space.

Following the mentioned methods in Section III-G, a series of self-supervised Transformers is also extended to 3-D spaces [113], [114], [115]. Specifically, Point-BERT [113] and Point-MAE [114] directly transfer the previous works [70], [71] to point clouds, while MaskPoint [115] changes the generative training scheme by using a contrastive decoder as similar as DINO (2022) [92] for binary noise/part classification. Based on large experiments, we can conclude that such generative/contrastive self-training methods empower visual Transformers to be valid in either images or points.

B. Cognition Mapping

Given rich representation features, how to directly map the instance/semantic cognition to the target outputs also arouse considerable interests. Different from 2-D images, the objects in 3-D scenes are independent and can be intuitively represented by a series of discrete surface points. To bridge the gap, some existing methods transfer domain knowledge into 2-D prevailing models. 3DETR [117] first extends Transformer detector to 3-D object detection via farthest point sampling and Fourier positional embeddings for object query initialization. Group-Free 3-D DETR [117] applies a more specified and stronger structure than [116]. In detail, it directly selects a set of candidate sample points from the extracted point clouds as the object queries and updates them in the decoder layer-by-layer iteratively. Moreover, the K-closed inside points are assigned positive and supervised by a binary objectiveness loss in both sampler and decoder heads. Sheng et al. [118] proposed a typical two-stage method that leverages a channelwise transformer 3-D detector (CT3D) to simultaneously aggregate proposal-aware embedding and channelwise context information for the point features within each proposal.

For monocular sensors, both MonoDTR [119] and MonoDETR [120] utilize an auxiliary depth supervision to estimate pseudo depth positional encodings (DPEs) during the training process. In MonoDETR [119], DPEs are first attached with the image features for Transformer encoder and then serve as the inputs of the DETR-like [30] decoder to initialize the object queries. In MonoDETR [120], both visual features and DPEs are first extracted by two different encoders parallelly and then interact with object queries via two successive cross-attention layers. Based on foreground depth supervision and narrow categorization interval, MonoDETR obtains the SoTA

result on the KITTI benchmark. DETR3D [121] introduces a multicamera 3-D object detection paradigm where both 2-D images and 3-D positions are associated by the camera transformation matrices and a set of 3-D object queries. TransFusion [122] further takes the advantages of both LiDAR points and RGB images by interacting with object queries through two Transformer decoder layers successively. More multisensory data fusion is introduced in Section VII-A.

C. Specific Processing

Limited by sensor resolution and view angle, point clouds are afflicted with incompletion, noise, and sparsity problems in real-world scenes. To this end, PoinTr [123] represents the original point cloud as a set of local point proxies and leverages a geometry-aware encoder–decoder Transformer to migrate the center point proxies toward incomplete points direction. SnowflakeNet [124] formulates the process of completing point clouds as a snowflake-like growth, which progressively generates child points from their parent points implemented by a pointwise splitting deconvolution strategy. A skip-Transformer for adjacent layers further refines the spatial-context features between parents and children to enhance their connection regions. Choe et al. [125] unified various generation tasks (e.g., denoising, completing, and super-resolution) into a point cloud reconstruction problem, hence termed PointRecon. Based on voxel hashing, it covers the absolute-scale local geometry and utilizes a PointTransformer-like [105] structure to aggregate each voxel (the query) with its neighbors (the value–key pair) for fine-grained conversion from the discrete voxel to a group of point sets. Moreover, an amplified positional encoding is adapted to the voxel local attention scheme, implemented by using a negative exponential function with L1-loss as weights for vanilla positional encodings. Notably, compared with masked generative self-training, the completion task directly generates a set of complete points without the explicit spatial prior of incomplete points.

VII. Transformer for Multisensory Data Stream

In the real world, multiple sensors are always used complementarily rather than a single one. To this end, recent works start to explore different fusing methods to cooperate multisensory data stream effectively. Compared with the typical CNNs, Transformer is naturally appropriate for multistream data fusion because of its nonspecific embedding and dynamically interactive attention mechanism. This section details these methods according to their data stream sources: homologous stream and heterologous stream.

A. Homologous Stream

Homologous stream is a set of multisensory data with similar inherent characteristics, such as multiview, multidimension, and multimodality visual stream data. They can be categorized into two groups: interactive fusion and transfer fusion, according to their fusion mechanism.

1) Interactive Fusion: The classical fusion pattern of CNN adopts a channel concatenation operation. However, the same positions from different modalities might be anisotropic, which is unsuitable for the translation-invariant bias of CNN. Instead, the spatial concatenation operation of Transformer enables different modalities to interact beyond the local restriction.

For the local interaction, MVT [126] spatially concatenates the patch embeddings from different views and strengthens their interaction via a modal-agnostic Transformer encoder. Considering the redundant features from different modalities, MVDeTr [127] projects each view of features onto the ground plane and extends the multiscale deformable attention [76] to a multiview design. TransFuser [128], COTR [129], and mmFormer [133] deploy a hybrid model. TransFuser models image and LiDAR inputs separately by using two different convolution backbones and links the intermediate feature maps via a Transformer encoder together with a residual connection. COTR shares the CNN backbone for each of view images and inputs the resulted features into a Transformer encoder block with a spatially expanded mesh-grid positional encoding. mmFormer exploits a modality-specific Transformer encoder for each sequence of MRI image and a modality-correlated Transformer encoder for multimodal modeling.

For the global interaction, Wang et al. [130] leveraged a shared backbone to extract the features for different views. Instead of pixelwise/patchwise concatenation in COTR [129], the extracted viewwise global features are spatially concatenated to perform view fusion within a Transformer. Considering the angular and position discrepancy across different camera views, TransformerFusion [132] first converts each view feature into an embedding vector with the intrinsics and extrinsics of their camera views. These embeddings are then fed into a global Transformer whose attention weights are used for a frame selection so as to compute efficiently. To unify the multisensory data in 3-D detection, FUTR3D [131] projects the object queries in the DETR-like decoder into a set of 3-D reference points. These points together with their related features are subsequently sampled from different modalities and spatially concatenated to update the object queries.

2) Transfer Fusion: Unlike the interactive fusion implemented by the Transformer encoder with self-attention, the other fusing form is more like a transfer learning from the source data to the target one via a cross-attention mechanism. For instance, Tulder et al. [134] inserted two cooperative cross-attention Transformers into the intermediate backbone features for bridging the unregistered multiview medical images. Instead of the pixelwise attention form, token-pixel cross attention is further developed to alleviate arduous computation. Long et al. [135] proposed an epipolar spatiotemporal Transformer for multiview image depth estimation. Given a single video containing a series of static multiview frames, the neighbor frames are first concatenated and the epipolar is then warped into the center camera space. The resulted frame volume finally serves as the source data to perform fusion with the center frame through a cross-attention block. With the spatially aligned data streams, DRT [136] first explicitly models the relation map between different data streams by using a convolution layer. The resulting maps are subsequently fed into a dual-path cross attention to build both local and global relationships parallelly, and thereby, it can collect more regional information for glaucoma diagnosis.

B. Heterologous Stream

Visual Transformers also perform excellently on heterologous data fusion, especially in visual-linguistic representation learning. Although different tasks may adopt different training schemes, such as supervised/self-supervised learning or compact/large-scale datasets, we categorize them into two representative groups only according to their cognitive forms: visual-linguistic pretraining including vision-language pretraining (VLP) and contrastive language-image pretraining (CLIP), and visual grounding such as phrase grounding (PG), referring expression comprehension (REC). For more details, see Table SII in the Supplementary Material.

1) Visual-Linguistic Pretraining: Due to limited annotated data, early VLP methods commonly rely on off-the-shelf object detector [200] and text encoder [5] to extract data-specific features for joint distribution learning. Given an image–text pair, an object detector pretrained on visual genome (VG) [201] first extracts a set of object-centric RoI features from the image. The RoI features serving as visual tokens are then merged with text embeddings for predefined task pretraining. Basically, these methods are grouped into dual- and single-stream fusion.

The dual-stream methods, including ViLBERT [138] and LXMERT [139], apply a vision-language cross-attention layer between two data-specific frameworks for multimodal transferring fusion. Concretely, ViLBERT [138] is pretrained through masked language modeling (MLM), masked region classification (MRC), and image text alignment (ITA) on conceptual captions (CCs) [202] with 3M image–text pairs. LXMERT [139] extends the pretraining datasets to a large-scale combination and further indicates that the pretrained task-specific (BERT [5]) weights initialization is harmful to the pretraining of multisensory data fusion.

VideoBERT [137] is the first single-stream VLP method, which clusters latent space features of each video frame as visual tokens and organizes the corresponding text embeddings via a captioning API. These features are then fed into a cross-modality self-attention layer for joint representation learning. Following [137], VisualBERT [140] extends such a single-stream framework for various image–text tasks and adds a segment embedding to distinguish between textual and visual tokens. VL-BERT [141] suggests that unmatched image–caption pairs over the ITA pretraining may decrease the accuracy of downstream tasks. Also, the authors further introduce both text-only corpus and unfrozen detector strategies for pretraining enhancement. Instead, such a "harmful" pretraining strategy is refuted by UNITER [142], and the authors deploy an optimal transport loss to explicitly build word-region alignment (WRA) at the instance level. To the same end, Oscar [143] uses shared linguistic semantic embeddings of a salient object class (called tag) as an anchor point to link both region and its paired words. Zhou et al. [144] proposed unified VLP to handle both generation and understanding tasks via a shared Transformer encoder–decoder with two customized attention masks. Without extra auxiliary training, unified VLP only adopts MLM during pretraining and attains superior results on visual question answering (VQA) [203] and visual captioning (VC) [204] tasks.

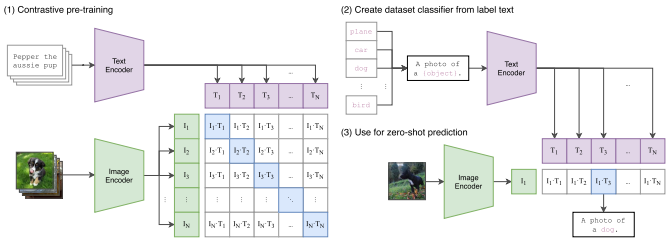

Fig. 12. Overview of CILP (from [147]).

However, these methods rely heavily on the visual extractor or predefined visual vocabulary, leading to a bottleneck of the VLP expressive upper bound. To address this issue, VinVL [146] develops an improved object detector for VLP pretraining on multiple large-scale dataset combinations. Instead of the object-centric RoI features, ViLT [145] initializes the interaction Transformer weights from a pretrained ViT and adopts whole word masking and image augmentation strategy for VLP pretraining. UniT [150] follows the architecture of DETR and applies a wide range of task for unified Transformer pretraining via different task-specific output heads simultaneously. SimVLM [151] adopts [39] to obtain image features and designs a prefix language modeling as a pretraining objective to generalize zero-shot image captioning.

Besides the conventional pretraining scheme with multitask supervision, another recent line has been developed for contrastive learning. The most representative work is CLIP [147]. Based on the 400M Internet image–text pairs datasets, both image and text encoders are jointly trained by a contrastive loss for ITA (Fig. 12). Notably, CLIP enables the pretrained model with a linear classifier to zero-shot transfer to the most visual downstream tasks efficiently by embedding the whole semantics of the objective datasets classes. Based on extensive experiments on over 30 existing CV tasks (e.g., classification and action recognition), CLIP attains superior results to classical supervised methods, demonstrating that such task-agnostic pretraining is also generalized well in the CV field. ALIGN [149] further expands a noisy dataset of over one billion image alt–text pairs rather than the elaborate filtering or postprocessing steps in CLIP [147].

Combining masked modeling and contrastive learning pretraining strategy, Data2Vec [152] proposes a self-distilled network treating the masked features as a type of data augmentation, whose structure is analogous to DINO (2021) [73]. By testing on different sensory benchmarks (voice, image, and language), it achieves competitive or better results compared with the existing self-supervised methods.

2) Visual Grounding: Compared with VLP, visual grounding has more concrete target signal supervision whose objective is to locate the target objects according to their corresponding descriptions. In the image space, modulated DETR (MDETR) [153] extends its previous work [30] to PG pretraining that locates and assigns the bounding box to each instance phrase in one description. Based on the proposed combined dataset from many existing ones, MDETR is first pretrained on the 1.3M aligned text–image pairs for PG and then fine-tuned on other downstream tasks. During pretraining, the image–text pair features are separately processed by two specific extractors and fed into a DETR-like Transformer for salient object localization.

Besides the box loss, two auxiliary losses are adopted to enforce the network to model an alignment between image feature and their corresponding phrase tokens. With the large-scale image–text pairs pretraining, MDETR can be easily generalized in few-shot learning, even on long-tail data. Different from MDETR [153] adding two auxiliary losses for box-phrase alignments, referring Transformer [156] directly initializes object queries with phrase-specific embeddings for PG, which explicitly reserves a one-to-one phrase assignment for final bounding box prediction. VGTR [155] reformulates the REC as a task for single salient object localization from the language features. In detail, a text-guided attention mechanism encapsulates both self-attention block and text–image cross-attention one to update the image features simultaneously. The resulted image features, which serve as the key–value pairs, interact with language queries when regressing bounding box coordinates in the decoder. Following ViT [29], TransVG [154] keeps the class token to aggregate the image and language features simultaneously for the mentioned object localization in REC. Pseudo-Q [157] focuses on REC for unsupervised learning, where a pseudo-query generation module based on a pretrained detector and a series of attributes&relationship generation algorithm is applied to generate a set of pseudo phrase descriptions and a query prompt is introduced to match feature proposals and phrase queries for REC adaptation.

In the 3-D spaces, LanguageRefer [158] redefines the multistream data reasoning as a language modeling problem, whose core idea is to omit point cloud features and infuse the predicted class embeddings together with a caption into a language model to get a binary prediction for object selection. Following the conventional two-stream methods, TransRefer3D [159] further enhances the relationship of the object features by using a cross-attention between asymmetric object relation maps and linguistic features. Considering the specific view for varied descriptions, Huang et al. [160] presented a multiview Transformer (MVT 2022) for 3-D visual grounding. Given a shared point cloud feature for each object, MVT first appends the converted bounding box coordinates to the shared objects in order to get specific view features. These multiview features are then fed into a stack of the Transformer decoders for text data fusion. Finally, the multiview features are merged by an order-independent aggregation function and converted to the grounding score. MVT achieves the SoTA performance on Nr3D and Sr3D datasets [205]. In the video space, a specific 3-D data (with temporal dimension), Yang et al. [161] proposed TubeDETR to address the problem of spatiotemporal video grounding (STVG). Concretely, a slow-fast encoder sparsely samples the frames and performs cross-modal self-attention between the sampled frames and the text features in the slow branch and aggregates the updated sample features into the full-frame features from fast branch via a broadcast operation. A learnable query attached with different time encodings, called time-specific queries in the decoder, is then predicted as either a time-aligned bounding box or "no object." It attains SoTA results on STVG leaderboards.

VIII. Conclusion and Discussion

This section briefly summaries the recent improvements in Section VIII-A, some critical issues disscusion in Section VIII-B, future research directions suggestion in Section VIII-C, and the final conclusion in Section VIII-D.

A. Summary of Recent Improvements

We briefly summarize the major performance improvements for three fundamental CV tasks as follows.

1) For classification, a deep hierarchical Transformer backbone is valid for decreasing the computational complexity [41] and avoiding the feature oversmooth [37], [42], [66], [67] in the deep layer. Meanwhile, the early stage convolution [39] is enough to capture the low-level features, which can significantly enhance the robustness and reduce the computational complexity in the shallow layer. Moreover, both the convolutional projection [54], [55] and the local attention mechanism [35], [44] can improve the locality of the visual Transformers. The former [56], [57] may also be a new approach to replace the positional encoding.
2) For detection, the Transformer necks benefit from the encoder–decoder structure with less computation than the encoder-only Transformer detector [88]. Thus, the decoder is necessary, but it requires more spatial prior [76], [80], [81], [82], [83], [85], [86] due to its slow convergence [87]. Furthermore, sparse attention [76] and scoring network [78], [79] for fore-grounding sampling are conducive to reducing the computational costs and accelerating the convergence of visual Transformers.
3) For segmentation, the encoder–decoder Transformer models may unify three segmentation subtasks into a mask prediction problem via a set of learnable mask embeddings [31], [104], [198]. This box-free approach has achieved the latest SoTA performance on multiple benchmarks [198]. Moreover, the specific hybrid task is cascaded with the model [101] of the box-based visual Transformers, which have demonstrated a higher performance for instance segmentation.
4) For 3-D visual recognition, the local hierarchical Transformer with a scoring network could efficiently extract features from the point clouds. Instead of the elaborate local design, the global modeling capability enables the Transformer to easily aggregate surface points. In addition, the visual Transformers can handle multisensory data in 3-D visual recognition, such as multiview and multidimension data.
5) The mainstream visual-linguistic pretraining have gradually focused on the alignments [147] or similarities [152] among different data streams in the latent space based on the large-scale noised datasets [149]. Another concern is to adapt the downstream visual tasks to the pretraining scheme to perform zero-short transferring [147].
6) The recent prevailing architecture for multisensory data fusion is the single-stream method, which spatially concatenates different data streams and performs interaction simultaneously. Based on the single-stream model, numerous recent works devote to finding a latent space to semantically align different data.

B. Discussion on Visual Transformers

Despite that the visual Transformer models are evolved significantly, the "essential" understanding remains insufficient.

Therefore, we will focus on reviewing some key issues for a deep and comprehensive understanding.

1) How Transformers Bridge the Gap Between Language and Vision: Transformers are initially designed for machine translation tasks [1], where each word of a sentence is taken as a basic unit representing the high-level semantics. These words can be embedded into a series of vector representations in the N-dimensional feature space. For visual tasks, each single pixel of an image is unable to carry semantic information, which is not full compliance with the feature embedding as done for the traditional NLP tasks. Therefore, the key for transferring such feature embeddings (i.e., word embedding) to image features and applying Transformer to various vision tasks is to build an image-to-vector transformation and maintain the image's characteristics effectively. For example, ViT [29] transforms an image into patch embeddings with multiple low-level information under strong slackness conditions. Also, its votarist [39], [58] leverages convolution to extract the low-level features and reduce the redundancy from patches.

2) Relationship Between Transformers, Self-Attention, and CNNs: From the perspective of CNNs, its inductive bias is mainly shown as locality, translation invariance, weight sharing, and sparse connection. Such a simple convolutional kernel can perform template matching efficiently in lower level semantic processing, but its upper bound tends to be lower than Transformers due to the excessive bias.

From the perspective of Transformers, as detailed in Sections III-B and III-D, attention layer can theoretically express any convolution when a sufficient number of heads are adopted [28]. Such fully attentional operation can combine both local-level attention and global-level attention and generate attention weights dynamically according to the feature relationships. Dong et al. [168] demonstrated that the self-attention layer manifests strong inductive bias toward "token uniformity" when it is trained on deep layers without short connection or FFNs. Yu et al. [206] also argued that such an elaborate attention mechanism can be replaced by a pooling operation readily. Therefore, it is concluded that Transformer must consist of two key components: a global token mixer (e.g., self-attention layer) aggregates the relationship of tokens and a positionwise FFN extracts the features from the inputs.

By comparison, the visual Transformer has a powerful global modeling capability, making it efficiently attend to high-level semantic features. CNNs can effectively process the low-level features [39], [58], enhance the locality of the visual Transformers [53], [81], and append the positional features via padding operations [56], [57], [173].

3) Double Edges of Visual Transformers: We conclude three double-edged properties of visual Transformers as follows. Global property enables Transformer to acquire capacious receptive fields and interact easily between various high-level semantic features, while it becomes inefficiency and debility during low-level processing because of quadratic computing and noised low-level features. Slack bias offers visual Transformer a higher upper bound than CNNs based on sufficient training data without sophistic assumptions but performs inferiority and slow convergence in small datasets [207]. Low pass is also a significant property of visual Transformer showing excellent robustness, whereas it is insensitive to low-level features (e.g., complicated textures and edges) compared with CNN. Accordingly, it is concluded that Transformers at a high-level stage play a vital role in various vision tasks.

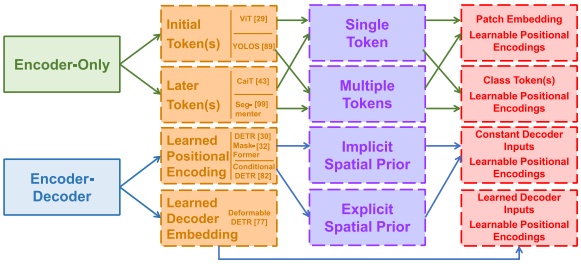

Fig. 13. Taxonomy of the learnable embedding.

4) Learnable Embeddings for Different Visual Tasks: Various learnable embeddings are designed to perform different visual tasks, such as class token, object query, and mask embedding. These learnable tokens are mainly adopted into two different Transformer patterns, i.e., encoder-only and encoder–decoder ones, as shown in Fig. 13. On the quantity level, the number of learned tokens depends on the target prediction. For example, the visual Transformers [29], [40] in the classification task adopt only one class token, and the DETR's votarist in detection [30], [81] and segmentation [198] tasks employ multiple learned queries. On the position level, encoder-only Transformers capitalize on the initial token(s) [29], [88] and later token(s) [42], [104], while the learned positional encoding [30], [81], [198] and the learned decoder input embedding [76] are applied to the encoder–decoder structure. Different from the vanilla ViT with initial class token, CaiT [42] observes that the later class token can reduce FLOPs of Transformer and improve the model performance slightly. Segmenter [104] also shows such strategy efficiency for the segmentation tasks. From the viewpoint of the encoder–decoder Transformer, the decoder input token is considered as a special case of the encoder-only Transformer with later token. It standardizes visual Transformers in the fields of detection [30] and segmentation [198] by using a small set of object queries (mask embeddings). By combing both later tokens and object queries (mask embeddings), the structure such as deformable DETR [76], which takes object queries and the learnable decoder embeddings (equivalent to the later tokens) as the inputs, may unify the learnable embeddings for different tasks into the encoder–decoder Transformer.

C. Future Research Directions

Visual Transformers have achieved significant progresses and obtained promising results. However, some key technologies are still insufficient to cope with complicated challenges in the CV fields. We point out some promising research directions for future investigation.

1) Set Prediction: Touvron et al. [40] found that multiple class tokens would converge consistently due to the same gradient from the loss function, whereas it does not emerge in dense prediction tasks [30], [198]. We conclude that their marked difference lies in the label assignment and the number of targets. Thus, it is natural to consider a set prediction design for the classification tasks, e.g., multiple

class tokens are aligned to mix-patches via set prediction, such as the data augmentation training strategy in LV-ViT [43]. Furthermore, the label assignment in the set prediction strategy leads to training instability during the early process, which degrades the accuracy of the final results. Redesigning the label assignments and set prediction losses may be helpful for the detection frameworks.

2) Self-Supervised Learning: Self-supervised pretraining of Transformers has standardized the NLP field and obtained tremendous successes in various applications [2], [5]. Because of the popularity of self-supervision paradigms in the CV field, the convolutional Siamese networks employ contrastive learning to perform self-supervised pretraining, which differs from the masked autoencoders used in the NLP field. Recently, some studies have tried to design self-supervised visual Transformers to bridge the discrepancy of pretraining methodology between vision and language. Most of them inherit the masked autoencoders in the NLP field or contrastive learning schemes in the CV field. There is no specific supervised method for the visual Transformers, but it has revolutionized the NLP tasks such as GPT-3. As described in Section VIII-B4, the encoder–decoder structure may unify the visual tasks by learning the decoder embedding and the positional encoding jointly. Thus, it is worth of further investigating the encoder–decoder Transformers for self-supervised learning.

D. Conclusion

Since ViT demonstrated its effectiveness for the CV tasks, the visual Transformers have received considerable attention and undermined the dominant of CNNs in the CV field. In this article, we have comprehensively reviewed more than 100 of visual Transformer models that have been successively applied to various vision tasks (i.e., classification, detection, and segmentation) and data streams (e.g., images, point clouds, image–text pairs, and other multiple data streams). For each vision task and data stream, a specific taxonomy is proposed to organize the recently developed visual Transformers and their performances are further evaluated over various prevailing benchmarks. From our integrative analysis and systematic comparison of all these existing methods, a summary of remarkable improvements is provided in this article, four essential issues for the visual Transformers are also discussed, and several potential research directions are further suggested for future investigation. We do expect that this review article can help readers have better understandings of various visual Transformers before they decide to perform deep explorations.

ACKNOWLEDGMENT

This work was done at the AI Lab, Lenovo Research, Beijing, China.

REFERENCES

[1] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[2] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, Tech. Rep., 2018.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, Tech. Rep., 2019.

[4] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, 2020, pp. 1877–1901.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2018, pp. 4171–4186.

[6] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[7] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. ICLR*, 2020, pp. 1–17.

[8] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. NeurIPS*, 2019, pp. 5753–5763.

[9] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[12] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[13] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021.

[14] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE CVPR*, Jun. 2018, pp. 7794–7803.

[15] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[16] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF CVPR*, Jun. 2018, pp. 7132–7141.

[18] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[19] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF CVPR*, Jun. 2020, pp. 11534–11542.

[20] N. Parmar et al., "Image transformer," in *Proc. ICML*, 2018, pp. 4055–4064.

[21] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE/CVF CVPR*, Jun. 2018, pp. 3588–3597.

[22] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 3464–3473.

[23] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. ICCV*, 2019, pp. 3286–3295.

[24] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Proc. NeurIPS*, 2019, pp. 68–80.

[25] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. IEEE/CVF CVPR*, Jun. 2020, pp. 10076–10085.

[26] Z. Zheng, G. An, D. Wu, and Q. Ruan, "Global and local knowledge-aware attention network for action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 334–347, Jan. 2021.

[27] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 12894–12904.

[28] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," in *Proc. ICLR*, 2020, pp. 1–18.

[29] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021, pp. 1–16.

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, 2020, pp. 213–229.

[31] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "MaX-DeepLab: End-to-end panoptic segmentation with mask transformers," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 5463–5474.

[32] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. NeurIPS*, 2021, pp. 17864–17875.

[33] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. CVPR*, 2021, pp. 8126–8135.

[34] Y. Jiang, S. Chang, and Z. Wang, "TransGAN: Two pure transformers can make one strong GAN, and that can scale up," 2021, *arXiv:2102.07074*.

[35] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted Windows," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 10012–10022.

[36] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 22–31.

[37] D. Zhou et al., "Refiner: Refining self-attention for vision transformers," 2021, *arXiv:2106.03714*.

[38] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," 2021, *arXiv:2106.04560*.

[39] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. NeurIPS*, 2021, pp. 3965–3977.

[40] H. Touvron, M. Cord, D. Matthijs, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. ICLR*, 2021, pp. 10347–10357.

[41] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 568–578.

[42] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 32–42.

[43] Z.-H. Jiang et al., "All tokens matter: Token labeling for training better vision transformers," in *Proc. NeurIPS*, 2021, pp. 18590–18602.

[44] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, "VOLO: Vision outlooker for visual recognition," 2021, *arXiv:2106.13112*.

[45] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–28, Apr. 2022.

[46] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–41, Jan. 2022.

[47] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[48] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, Oct. 2022.

[49] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, 2014, pp. 3104–3112.

[50] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[51] B. Wu et al., "Visual transformers: Token-based image representation and processing for computer vision," 2020, *arXiv:2006.03677*.

[52] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 16519–16529.

[53] S. D'Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun, "ConViT: Improving vision transformers with soft convolutional inductive biases," in *Proc. ICLR*, 2021, pp. 2286–2296.

[54] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating convolution designs into visual transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 579–588.

[55] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "LocalViT: Bringing locality to vision transformers," 2021, *arXiv:2104.05707*.

[56] X. Chu et al., "Conditional positional encodings for vision transformers," 2021, *arXiv:2102.10882*.

[57] Q. Zhang and Y.-B. Yang, "ResT: An efficient transformer for visual recognition," in *Proc. NeurIPS*, 2021, pp. 15475–15485.

[58] T. Xiao, P. Dollár, M. Singh, E. Mintun, T. Darrell, and R. Girshick, "Early convolutions help transformers see better," in *Proc. NeruIPS*, 2021, pp. 30392–30400.

[59] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. NeurIPS*, 2021, pp. 15908–15919.

[60] X. Chu et al., "Twins: Revisiting the design of spatial attention in vision transformers," in *Proc. NeurIPS*, 2021, pp. 9355–9366.

[61] P. Zhang et al., "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 2998–3008.

[62] J. Yang et al., "Focal self-attention for local-global interactions in vision transformers," 2021, *arXiv:2107.00641*.

[63] L. Yuan et al., "Tokens-to-Token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 558–567.

[64] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 11936–11945.

[65] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," 2021, *arXiv:2106.13797*.

[66] D. Zhou et al., "DeepViT: Towards deeper vision transformer," 2021, *arXiv:2103.11886*.

[67] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "Vision transformers with patch diversification," 2021, *arXiv:2104.12753*.

[68] M. Chen et al., "Generative pretraining from pixels," in *Proc. ICML*, 2020, pp. 1691–1703.

[69] Z. Li et al., "MST: Masked self-supervised transformer for visual representation," in *Proc. NeurIPS*, 2021, pp. 13165–13176.

[70] H. Bao, L. Dong, and F. Wei, "BEiT: BERT pre-training of image transformers," in *Proc. ICLR*, 2021, pp. 1–18.

[71] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 16000–16009.

[72] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 9640–9649.

[73] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 9650–9660.

[74] Z. Xie et al., "Self-supervised learning with Swin transformers," 2021, *arXiv:2105.04553*.

[75] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton, "Pix2seq: A language modeling framework for object detection," in *Proc. ICLR*, 2021, pp. 1–17.

[76] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. ICLR*, 2021, pp. 1–16.

[77] M. Zheng et al., "End-to-End object detection with adaptive clustering transformer," 2020, *arXiv:2011.09315*.

[78] T. Wang, L. Yuan, Y. Chen, J. Feng, and S. Yan, "PnP-DETR: Towards efficient visual analysis with transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 4661–4670.

[79] B. Roh, J. Shin, W. Shin, and S. Kim, "Sparse DETR: Efficient end-to-end object detection with learnable sparsity," in *Proc. ICLR*, 2021, pp. 1–23.

[80] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of DETR with spatially modulated co-attention," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 3621–3630.

[81] D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 3651–3660.

[82] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based detector," in *Proc. AAAI*, 2022, pp. 2567–2575.

[83] S. Liu et al., "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *Proc. ICLR*, 2021, pp. 1–19.

[84] Y. Liu et al., "SAP-DETR: Bridging the gap between salient points and queries-based transformer detector for fast model convergency," 2022, *arXiv:2211.02006*.

[85] Z. Yao, J. Ai, B. Li, and C. Zhang, "Efficient DETR: Improving end-to-end object detector with dense prior," 2021, *arXiv:2104.01318*.

[86] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, and L. Zhang, "Dynamic DETR: End-to-end object detection with dynamic attention," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 2988–2997.

[87] Z. Sun, S. Cao, Y. Yang, and K. Kitani, "Rethinking transformer-based set prediction for object detection," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 3611–3620.

[88] Y. Fang et al., "You only look at one sequence: Rethinking transformer in vision through object detection," in *Proc. NeurIPS*, 2021, pp. 26183–26197.

[89] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 1601–1610.

[90] W. Wang, Y. Cao, J. Zhang, and D. Tao, "FP-DETR: Detection transformer advanced by fully pre-training," in *Proc. ICLR*, 2021, pp. 1–14.

[91] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query denoising," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 13619–13627.

[92] H. Zhang et al., "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.

[93] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun, "Feature pyramid transformer," in *Proc. ECCV*, 2020, pp. 323–339.

[94] Y. Yuan et al., "HRFormer: High-resolution vision transformer for dense predict," in *Proc. NeurIPS*, 2021, pp. 7281–7293.

[95] J. Gu et al., "HRViT: Multi-scale high-resolution vision transformer," 2021, *arXiv:2111.01236*.

[96] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 6881–6890.

[97] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[98] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. NeurIPS*, 2021, pp. 12077–12090.

[99] T. Prangemeier, C. Reich, and H. Koeppl, "Attention-based transformers for instance segmentation of cells in microstructures," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2020, pp. 700–707.

[100] Y. Wang et al., "End-to-end video instance segmentation with transformers," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 8741–8750.

[101] Y. Fang et al., "Instances as queries," in *Proc. ICCV*, 2021, pp. 6910–6919.

[102] J. Hu et al., "ISTR: End-to-end instance segmentation with transformers," 2021, *arXiv:2105.00637*.

[103] B. Dong, F. Zeng, T. Wang, X. Zhang, and Y. Wei, "SOLQ: Segmenting objects by learning queries," in *Proc. NeurIPS*, 2021, pp. 21898–21909.

[104] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 7262–7272.

[105] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 16259–16268.

[106] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, 2021.

[107] D. Lu, Q. Xie, K. Gao, L. Xu, and J. Li, "3DCTN: 3D convolution-transformer network for point cloud classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24854–24865, Dec. 2022.

[108] C. Park, Y. Jeong, M. Cho, and J. Park, "Fast point transformer," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 16949–16958.

[109] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with pointformer," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 7463–7472.

[110] L. Fan et al., "Embracing single stride 3D object detector with sparse transformer," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 8458–8468.

[111] J. Mao et al., "Voxel transformer for 3D object detection," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 3144–3153.

[112] C. He, R. Li, S. Li, and L. Zhang, "Voxel set transformer: A set-to-set approach to 3D object detection from point clouds," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 8417–8427.

[113] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-training 3D point cloud transformers with masked point modeling," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 19313–19322.

[114] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," 2022, *arXiv:2203.06604*.

[115] H. Liu, M. Cai, and Y. Jae Lee, "Masked discrimination for self-supervised learning on point clouds," 2022, *arXiv:2203.11183*.

[116] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 2906–2917.

[117] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3D object detection via transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 2949–2958.

[118] H. Shenga et al., "Improving 3D object detection with channel-wise transformer," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 2743–2752.

[119] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "MonoDTR: Monocular 3D object detection with depth-aware transformer," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 4012–4021.

[120] R. Zhang et al., "MonoDETR: Depth-guided transformer for monocular 3D object detection," 2022, *arXiv:2203.13310*.

[121] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3D object detection from multi-view images via 3D-to-2D queries," in *Proc. 5th Conf. Robot Learn.*, 2022, pp. 180–191.

[122] X. Bai et al., "TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 1090–1099.

[123] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Proc. ICCV*, 2021, pp. 12478–12487.

[124] P. Xiang et al., "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 5479–5489.

[125] J. Choe, B. Joung, F. Rameau, J. Park, and I. So Kweon, "Deep point cloud reconstruction," 2021, *arXiv:2111.11704*.

[126] S. Chen, T. Yu, and P. Li, "MVT: Multi-view vision transformer for 3D object recognition," 2021, *arXiv:2110.13083*.

[127] Y. Hou and L. Zheng, "Multiview detection with shadow transformer (and view-coherent data augmentation)," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1673–1682.

[128] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7077–7087.

[129] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi, "COTR: Correspondence transformer for matching across images," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 6207–6217.

[130] D. Wang et al., "Multi-view 3D reconstruction with transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 5722–5731.

[131] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "FUTR3D: A unified sensor fusion framework for 3D detection," 2022, *arXiv:2203.10642*.

[132] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Nießner, "TransformerFusion: Monocular RGB scene reconstruction using transformers," in *Proc. NeurIPS*, 2021, pp. 1403–1414.

[133] Y. Zhang et al., "mmFormer: Multimodal medical transformer for incomplete multimodal learning of brain tumor segmentation," in *Proc. MICCAI*, 2022, pp. 107–117.

[134] G. V. Tulder, Y. Tong, and E. Marchiori, "Multi-view analysis of unregistered medical images using cross-view transformers," in *Proc. MICCAI*, 2021, pp. 104–113.

[135] X. Long, L. Liu, W. Li, C. Theobalt, and W. Wang, "Multi-view depth estimation using epipolar spatio-temporal networks," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 8258–8267.

[136] D. Song et al., "Deep relation transformer for diagnosing glaucoma with optical coherence tomography and visual field function," *IEEE Trans. Med. Imag.*, vol. 40, no. 9, pp. 2392–2402, Sep. 2021.

[137] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "VideoBERT: A joint model for video and language representation learning," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 7464–7473.

[138] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proc. NeurIPS*, 2019, pp. 13–23.

[139] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5100–5111.

[140] L. Harold Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "VisualBERT: A simple and performant baseline for vision and language," 2019, *arXiv:1908.03557*.

[141] W. Su et al., "VL-BERT: Pre-training of generic visual-linguistic representations," 2019, *arXiv:1908.08530*.

[142] Y.-C. Chen et al., "UNITER: Universal image-text representation learning," in *Proc. ECCV*, 2020, pp. 104–120.

[143] X. Li et al., "OSCAR: Object-semantics aligned pre-training for vision-language tasks," in *Proc. ECCV*, 2020, pp. 121–137.

[144] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and VQA," in *Proc. AAAI*, 2020, pp. 13041–13049.

[145] W. Kim, B. Son, and I. Kim, "ViLT: Vision-and-language transformer without convolution or region supervision," in *Proc. ICML*, 2021, pp. 5583–5594.

[146] P. Zhang et al., "VinVL: Revisiting visual representations in vision-language models," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 5579–5588.

[147] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. ICML*, 2021, pp. 8748–8763.

[148] A. Ramesh et al., "Zero-shot text-to-image generation," in *Proc. ICML*, 2021, pp. 8821–8831.

[149] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. ICML*, 2021, pp. 4904–4916.

[150] R. Hu and A. Singh, "UniT: Multimodal multitask learning with a unified transformer," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 1439–1449.

[151] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "SimVLM: Simple visual language model pretraining with weak supervision," in *Proc. ICLR*, 2021, pp. 1–17.

[152] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," 2022, *arXiv:2202.03555*.

[153] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "MDETR—Modulated detection for end-to-end multi-modal understanding," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 1780–1790.

[154] J. Deng, Z. Yang, T. Chen, W. Zhou, and H. Li, "TransVG: End-to-end visual grounding with transformers," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 1769–1779.

[155] Y. Du, Z. Fu, Q. Liu, and Y. Wang, "Visual grounding with transformers," in *Proc. ICME*, 2022, pp. 1–6.

[156] M. Li and L. Sigal, "Referring transformer: A one-step approach to multi-task visual grounding," in *Proc. NeurIPS*, 2021, pp. 19652–19664.

[157] H. Jiang, Y. Lin, D. Han, S. Song, and G. Huang, "Pseudo-Q: Generating pseudo language queries for visual grounding," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 15513–15523.

[158] J. Roh, K. Desingh, A. Farhadi, and D. Fox, "LanguageRefer: Spatial-language model for 3D visual grounding," in *Proc. 5th Conf. Robot Learn.*, 2022, pp. 1046–1056.

[159] D. He et al., "TransRefer3D: Entity-and-relation aware transformer for fine-grained 3D visual grounding," in *Proc. 29th ACM Int. Conf. Multimedia (CMM)*, 2021, pp. 2344–2352.

[160] S. Huang, Y. Chen, J. Jia, and L. Wang, "Multi-view transformer for 3D visual grounding," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 15524–15533.

[161] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "TubeDETR: Spatio-temporal video grounding with transformers," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 16442–16453.

[162] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[163] P. P. Brahma, D. Wu, and Y. She, "Why deep learning works: A manifold disentanglement perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 1997–2008, Oct. 2016.

[164] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A^2-Nets: Double attention networks," in *Proc. NeurIPS*, 2018, pp. 352–361.

[165] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.

[166] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE ICCV*, Oct. 2017, pp. 843–852.

[167] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[168] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," in *Proc. ICLR*, 2021, pp. 2793–2803.

[169] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 464–468.

[170] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*.

[171] S. Abnar, M. Dehghani, and W. Zuidema, "Transferring inductive biases through knowledge distillation," 2020, *arXiv:2006.00555*.

[172] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF CVPR*, Jun. 2018, pp. 4510–4520.

[173] A. Islam, S. Jia, and N. D. B. Bruce, "How much position information do convolutional neural networks encode?" in *Proc. ICLR*, 2020, pp. 1–11.

[174] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 7083–7093.

[175] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1587–1597, May 2018.

[176] J. Gao, D. He, X. Tan, T. Qin, L. Wang, and T.-Y. Liu, "Representation degeneration problem in training natural language generation models," in *Proc. ICLR*, 2019, pp. 1–14.

[177] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 6023–6032.

[178] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. NeurIPS*, 2020, pp. 9912–9924.

[179] C.-F.-R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF ICCV*, Oct. 2021, pp. 357–366.

[180] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[181] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE CVPR*, Jun. 2016, pp. 779–788.

[182] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE ICCV*, Oct. 2017, pp. 2980–2988.

[183] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[184] J. Dai et al., "Deformable convolutional networks," in *Proc. ICCV*, 2017, pp. 764–773.

[185] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE CVPR*, Jul. 2017, pp. 2117–2125.

[186] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 9627–9636.

[187] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, 2017, pp. 2961–2969.

[188] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE CVPR*, Jun. 2018, pp. 6154–6162.

[189] P. Sun et al., "What makes for end-to-end object detection?" in *Proc. ICML*, 2021, pp. 9934–9944.

[190] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF CVPR*, Jun. 2019, pp. 5693–5703.

[191] L. C. Chen, G. Papandreou, and I. Kokkinos, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Jun. 2016.

[192] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[193] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF CVPR*, Jun. 2021, pp. 14454–14463.

[194] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. ECCV*, 2018, pp. 418–434.

[195] M. Chen et al., "Searching the search space of vision transformer," in *Proc. NeurIPS*, 2021, pp. 1–13.

[196] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "BlendMask: Top-down meets bottom-up for instance segmentation," in *Proc. IEEE/CVF CVPR*, Jun. 2020, pp. 8573–8581.

[197] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic and fast instance segmentation," in *Proc. NeurIPS*, 2020, pp. 17721–17732.

[198] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16×16 words: Dynamic transformers for efficient image recognition," in *Proc. NeurIPS*, 2021, pp. 11960–11973.

[199] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NeurIPS*, 2017, pp. 1–10.

[200] P. Anderson et al., "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE/CVF CVPR*, Jun. 2018, pp. 6077–6086.

[201] R. Krishna et al., "Visual Genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

[202] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2556–2565.

[203] S. Antol et al., "VQA: Visual question answering," in *Proc. IEEE ICCV*, Dec. 2015, pp. 2425–2433.

[204] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE CVPR*, Jun. 2015, pp. 3156–3164.

[205] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas, "ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes," in *Proc. ECCV*, 2020, pp. 422–440.

[206] W. Yu et al., "MetaFormer is actually what you need for vision," in *Proc. IEEE/CVF CVPR*, Jun. 2022, pp. 10819–10829.

[207] N. Park and S. Kim, "How do vision transformers work?" in *Proc. ICLR*, 2021, pp. 1–26.